AFRL-IF-RS-TR-2007-89
In-House Final Technical Report
March 2007

THE MATRIX PENCIL AND ITS APPLICATIONS TO SPEECH PROCESSING

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Rome Research Site Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-IF-RS-TR-2007-89 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
ANDREW J. NOGA, Acting Chief Multi-Sensor Exploitation Branch	JOSEPH CAMERA, Chief Information & Intelligence Exploitation Division Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
MAR 2007	Final	Feb 04 – Dec 06

4. TITLE AND SUBTITLE

THE MATRIX PENCIL AND ITS APPLICATIONS TO SPEECH PROCESSING

5a. CONTRACT NUMBER
In-House

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62702F

6. AUTHOR(S)

Darren H. Haddad and Andrew J. Noga

5d. PROJECT NUMBER
459E

5e. TASK NUMBER
IH

5f. WORK UNIT NUMBER
SP

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
AFRL/IFEC
525 Brooks Rd
Rome NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFEC
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2007-89

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 07-136

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Matrix Pencils facilitate the study of differential equations resulting from oscillating systems. Certain problems in linear ordinary differential equations, such as speech processing, can be represented as the problem of finding a canonical pencil strictly equivalent to a given pencil. It was originally applied by the radar community to phased array radar for signal directional finding applications. The Matrix Pencil (MP) algorithm is a direct data approach, and is a nonstochastic method. This approach has many benefits over a statistical approach. One benefit allows the user to approximate the error of the reconstructed signal without reconstructing the signal. Second, it takes less time and less computational power to execute the algorithm. Third, the matrix pencil approach has a lower variance of the estimates of the parameters of interest than a statistical approach such as traditional Linear Prediction Coding (LPC). Speech processing has many applications which directly assist in the advancement of technology. These technologies utilize speech tools that include, but are not limited to speech compression, speech enhancement, speech recovery, pitch estimation, and co-channel interference reduction. However, the speech processing community has not grasped the power of the MP algorithm, which will likely make a significant leap forward in improving these speech processing tools.

15. SUBJECT TERMS
Matrix Pencil, Speech Processing, Speech Compression, Speech Enhancement, Filtering, Pitch Estimation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Darren M. Haddad
U	U	U	UL	152	19b. TELEPHONE NUMBER (*Include area code*)

Table of Contents

List of Figures ..... **iv**

List of Tables ..... **viii**

List of Abbreviations ..... **ix**

Introduction **1**

1.1 Matrix Pencil Overview 1

1.2 Speech Processing 5

1.2.1 Speech Production 5

1.2.2 Speech Modeling 7

1.2.3 Short Time Characteristics of Speech 9

1.2.4 Speech Processing Applications 13

1.3 Dissertation Outline and Overview 15

Matrix Pencil **17**

2.1 The Theory of the Matrix Pencil 17

2.2 Model for the Sum of Complex Exponentials 20

2.3 MP Algorithm 24

2.3.1 Data Approach 24

2.3.2 Eigenvector Approach 29

2.4 MP Computational Stability with Speech Processing 31

2.4.1 Numerical Considerations 31

2.4.2 Condition, Stability, and Location of the Poles 33

2.4.2.1 Ill-Condition Examples and Observations 35

2.4.2.2 Ill-Conditioning Handling: Backward Method 42
2.4.2.3 Ill-Condition Handling: Triangular Window Weighted Algorithm 46
2.4.3 Double Poles and Close Proximity Singular Values 47
2.5 Reconstruction Error 48
2.5.1 Relative Error 49
2.5.2 Results of Low Rank Error Modeling 53
Speech Compression Using the Matrix Pencil 63
3.1 Introduction 63
3.2 MP Speech Compression Algorithm 64
3.3 MP Speech Compression Results 65
3.4 Conclusion 69
Band Focus Matrix Pencil Algorithm 71
4.1 Introduction 71
4.2 Stationary Tonal Detection 72
4.2.1 Strong Tone Detection 73
4.2.2 Moderate Tone Detection 76
4.2.3 Weak Tone Detection 78
4.2.4 Stationary Tone Detection Conclusion 81
4.3 Tone Removal 82
4.3.1 Notch Filter Baseline 83
4.3.2 Band Focus Matrix Pencil Temporal Subtraction 84
4.3.3 Band Focus Matrix Pencil Temporal Reconstruction 87
4.3.3.1 Cascade BFMP-TR 89

4.3.4 Single Tone, FM Signal, AM Signal, and Multi-Tone Testing 91
4.3.4.1 SID and RMSE in the presences of a Stationary Interfering Tone 93
4.3.4.2 SID in the presences of Multiple Amplitude Level Interfering Tone 96
4.3.4.3 SID and RMSE in the presences of a FM Interfering Signal 98
4.3.4.4 SID and RMSE in the presences of an AM interfering signal 103
4.3.4.5 SID and RMSE in the presences of a 60 Hz Interfering Tone and its Odd Harmonics 106
4.4 Conclusion 109
Pitch Tracking using the Generalized Harmonicity Indicator 111
5.1 Introduction 111
5.2 Pre Processing 112
5.3 Voiced/ Unvoiced Detection 114
5.4 Pitch Estimation Algorithm 115
5.5 Results 119
5.6 Conclusions 123
Conclusions 126
6.1 Conclusions and Future Work 126
Appendix A 132
Reference 134

List of Figures

Figure 1.1: Cross section of the vocal tract. 6

Figure 1.2: Model of the vocal tract. 7

Figure 1.3: Common window functions. 10

Figure 1.4: Fourier Transform (log Magnitude) of 512 sample length for the Hanning, Hamming, Blackman, Bartlett, and Rectangular window functions. 10

Figure 2.1: A 20 ms frame, using 6 poles for signal reconstruction out of a possible 79 poles. 36

Figure 2.2: A z-plane plot of the poles for the corresponding plot of Figure 2.1. 37

Figure 2.3: A 20 ms frame using 27 poles for signal reconstruction out of a possible 79 poles (Ill-conditioned). 37

Figure 2.4: A z-plane plot of the poles for the corresponding plot of Figure 2.3. 38

Figure 2.5: A 20 ms frame, using 28 poles for signal reconstruction out of a possible 79 poles. 39

Figure 2.6: A z-plane plot of the poles for the corresponding plot of Figure 2.5. 40

Figure 2.7: A 20 ms frame, using 29 poles for signal reconstruction out of a possible 79 poles (Ill-conditioned). 40

Figure 2.8: A z-plane plot of the poles for the corresponding plot of Figure 2.7. 41

Figure 2.9: The mean square error between the reconstructed and the original signal for a frame of data reconstructed at a different number of poles. 41

Figure 2.10: A 20 ms frame, in backward mode, using 29 poles for signal reconstruction out of a possible 79 poles. 43

Figure 2.11: A z-plane plot of the poles for the corresponding plot of Figure 2.10. 43

Figure 2.12: The MSE of a reconstructed frame of audio using 3 methods. 45

Figure 2.13: The MSE of a reconstructed frame of audio using 3 methods. 45

Figure 2.14: Triangular Filtering with 50% overlap. 47

Figure 2.15: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. 55

Figure 2.16: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. 55

Figure 2.17: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. 56

Figure 2.18: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. 56

Figure 2.19: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. 58

Figure 2.20: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. 58

Figure 2.21: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. 59

Figure 2.22: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. 59

Figure 2.23: Comparing the error ratios using the reconstructed signal and the original signal as a baseline, for a specific frame of speech. 60

Figure 2.24: Comparing the error ratios using the reconstructed signal and the original signal as a baseline, for a specific frame of speech. 61

Figure 2.25: Comparing the error ratios using the reconstructed signal and the original signal as a baseline, for a specific frame of speech. 61

Figure 2.26: Comparing the error ratios using the reconstructed signal and the original signal as a baseline, for a specific frame of speech. 62

Figure 2.27: Comparing the error ratios using the reconstructed signal and the original signal as a baseline, for a specific frame of speech. 62

Figure 4.1: Singular Value plot, speech vs. speech + tone mixed. 73

Figure 4.2: PSD of speech signal, tone signal, and speech + tone signal. 74

Figure 4.3: PSD comparison of original tone and reconstructed tone signal. 75

Figure 4.4: Time domain plot of the reconstructed tone signal. 75

Figure 4.5: Singular Value plot, speech vs. speech + tone mixed. 76

Figure 4.6: Power spectrum density comparison of a speech frame. 77

Figure 4.7: Time domain of reconstructed signal. 78

Figure 4.8: Singular Value plot, speech vs. speech + tone mixed. 79

Figure 4.9: Power spectrum density comparison of a speech frame. 79

Figure 4.10: Power spectrum density comparison of a speech frame. 80

Figure 4.11: Power spectrum density comparison of a speech frame. 80

Figure 4.12: BFMP-TS algorithm: Matrix Pencil combination with band pass filter to subtract the interfering tone. 85

Figure 4.13: Spectrogram of the original audio signal. 85

Figure 4.14: Spectrogram of the original audio signal mixed with a 400 Hz tone. 86

Figure 4.15: Spectrogram of an audio signal after BFMP-TS. 86

Figure 4.16: Spectrogram of an audio signal after a 15% notch (15% of the tone frequency, 400 Hz. 86

Figure 4.17: BFMP-TR algorithm: Matrix Pencil combination with band pass filter, to reconstruction speech while removing the interfering tone. 88

Figure 4.18: Spectrogram of an audio signal after BFMP-TR. 88

Figure 4.19: Block Diagram of the Cascade BFMP-TR. 91

Figure 4.20: SID results after a 400 Hz tone was removed using multiple removal techniques. 95

Figure 4.21: RMSE results after a 400 Hz tone was removed using multiple removal techniques. 95

Figure 4.22: Speaker Identification on the removal of multiple tone amplitude levels. 97

Figure 4.23: Zoom in version of Figure 4.22. 98

Figure 4.24: SID results after a 400 Hz FM signal (Δf=20Hz, f_m=2Hz) was removed using multiple removal techniques. 100

Figure 4.25: RMSE results after a 400 Hz FM signal ($\Delta f = 20Hz$, $f_m=2Hz$) was removed using multiple removal techniques. 101

Figure 4.26: SID results after a 400 Hz FM signal ($\Delta f = 10Hz$, $f_m=1Hz$) was removed using multiple removal techniques. 101

Figure 4.27: RMSE results after a 400 Hz FM signal ($\Delta f = 10Hz$, $f_m=1Hz$) was removed using multiple removal techniques. 102

Figure 4.28: SID results after a 400 Hz FM tone ($\Delta f = 5Hz$, $f_m=1Hz$) was removed using multiple removal techniques. 102

Figure 4.29: RMSE results after a 400 Hz FM signal ($\Delta f = 5Hz$, $f_m=1Hz$) was removed using multiple removal techniques. 103

Figure 4.30: Speaker Id results after a 400 Hz AM tone ($f_m = 25$ Hz, modulation index = .5) was removed using multiple removal techniques. 104

Figure 4.31: RMSE results after a 400 Hz tone AM tone ($f_m = 25$ Hz, modulation index = .5) was removed using multiple removal techniques. 105

Figure 4.32: Spectrogram of the original signal. 107

Figure 4.33: Spectrogram of the contaminated signal. 107

Figure 4.34: Spectrogram of the enhanced signal using the Notch Filter @ 7%. 107

Figure 4.35: Spectrogram of the enhanced signal using the BFMP-TR @ 5%. 108

Figure 4.36: SID results after removing multiple harmonic tones (Baseline 96.8%).108

Figure 4.37: RMSE results after removing multiple harmonic tones. 109

Figure 5.1: The Generalized Harmonicity Indicator. 116

List of Tables

Table 1.1: The different and common techniques used in each of the estimation methods. 4

Table 2.1: Comparison of different frame size vs. processing time for a similar number of poles (speech file =26000 samples @ 8kHz). 32

Table 2.2: Comparison of frame size to maximum pole radius. 35

Table 3.1: SNR and Compression Ratio of MP Speech Coders. 67

Table 3.2: SNR and Compression Ratio of MP and other Speech Coders. 68

Table 5.1: Fundamental estimation evaluation for male speech (top) and female speech (bottom). 120

Table 5.2: Fundamental estimation evaluation for male speech (top) and female speech (bottom) with varying thresholds for the SVD voiced unvoiced detector. 122

Table 5.3: MP GHI fundamental estimation evaluation during perfect detection. 123

List of Abbreviations

absolute deviation mean	adm
Analog to Digital	A/D
Autoregressive	AR
Band Focus Matrix Pencil	BFMP
Band Focus Matrix Pencil Temporal Reconstruction	BFMP-TR
Band Focus Matrix Pencil Temporal Subtraction	BFMP-TS
Complex (non-decaying) sinusoid	Cisoid
Digital Signal Processor	DSP
Discrete Fourier Transform	DFT
Enhanced Super Resolution Pitch Determinator	eSRPD
Estimation of Signal Parameters via Rotational Invariance Technique	ESPRIT
Exponential Decaying/Growing Sinusoids Model	ESM
Finite Impulse Response	FIR
floating point relative accuracy	eps
Generalized Harmonicity Indicator	GHI
kilo-bits per second	kbps
Linear Prediction	LP
Linear Prediction Coding	LPC
Linear Prediction Coder 10	LPC10
Matrix Pencil	MP
Mean Opinion Score	MOS

Mean Square Error	MSE
Moore-Penrose Pseudo-inverse	MPP
Power Spectral Density	PSD
population standard deviation	p.s.d.
Radio Frequency	RF
Root Mean Square Error	RMSE
Singular Value Decomposition	SVD
Speaker Identification	SID
Super Resolution Pitch Determinator	SRPD
Structured Total Least Norm	STLN
Total Least Square	TLS
Voice Over Internet Protocol	VOIP

Chapter 1

Introduction

1.1 Matrix Pencil Overview

In this work the Matrix Pencil (MP) algorithm is studied, in conjunction with speech processing. The MP algorithm is a technique, which estimates the parameters of a signal that is represented by an exponential decaying/growing sinusoids model (ESM). For speech processing modeling, ESM has shown to be a valuable tool over other models. This is due to the transient segments found in speech [1]. The MP algorithm is a non stochastic estimating technique that uses a snapshot of data to calculate its parameters. Directional of Arrival estimation has been the main application that has utilized the MP algorithm. The speech processing community has yet to determine all the benefits of this algorithm. The MP algorithm has the capability of providing a high resolution estimate of the signal's spectrum. With this advantage, the speech community can develop different methods to improve speech processing applications. Three of the applications that were studied in this dissertation are speech compression, speech enhancement, and pitch estimation.

Estimating the poles or frequencies of an ESM signal has been studied for many years. Many methods have been developed for estimating the parameters of an ESM signal. The concept of using singular value decomposition (SVD) techniques for estimating these types of signal became evident in the 1980's. One of the methods was

linear prediction (LP) using the eigenstructure [2] and [3]. In this method the authors used eigenvalue-eigenvector decomposition to replace the estimated correlation matrix in LP by a matrix of specific rank which is a least squares approximation to it. Another method called the Estimation of Signal Parameters via Rotational Invariance Technique (ESPRIT) is a high resolution signal parameter estimator [4] and [5]. The ESPRIT is a stochastic spectrum estimation technique. It uses the auto and cross covariance matrices to determine the poles. The ESPRIT models a non decaying sinusoid or a non decaying complex sinusoid (cisoid). The MP algorithm and the ESPRIT algorithm are similar with their technique of using a generalized eigenvalue approach, but that is the only similarity. The MP algorithm, a non stochastic estimation technique, is a direct data approach that uses an ESM. A variation of ESPRIT algorithm in [6] moves towards the MP algorithm. In this paper the author uses a snapshot of the data to determine the eigenvectors.

The concept of retrieving parameters of a sinusoid from noisy data, using the MP algorithm, initially came from [7]. The author indicates how the SVD possesses numerical properties for this type of problem. The first work of applying the MP algorithm to ESM signals was in [8,9,10,11, and12]. This research was done at Syracuse University. The work in these papers concentrated on ESM for the directional finding application. In [10 and 12], the authors compared the performance of the MP algorithm to the Prony method, Pisarenko method and the Pencil-of-Function method. It was determined that the MP estimate variance is better and closer to the Cramer Rao bound when comparing to these other methods.

Most of the MP research has centered on the direction finding application, until the author in [13] applied it to musical signals. In this paper, the author showed how the

MP algorithm can be used to model music signals. He concluded that the MP method is extremely accurate in the cases where other techniques, such as the Fourier transform, perform poorly. He also stated that the MP algorithm is very robust, regardless of its numerical complexity. A reason for the numerical complexity was due to the data not being windowed properly prior to processing. The windowing of the data is an important step in processing an audio signal with the MP algorithm, as will be discussed in chapter 2. The author never followed up with any additional research on this topic.

There has been limited amount of work with the MP algorithm and audio signals since the publication of [13]. In one publication [14], the author attempts to obtain prediction coefficients from the MP parameters using a linear prediction model. Another publication [15], the author uses the ESM and generates the poles using a MP approach. In that paper he proves that speech can be modeled using the MP approach. He also attempted a refinement to the MP algorithm using the Structured Total Least Norm (STLN) process. This refinement process attempts to give the best approximation to the original signal. He does not state how much more the refinement process improves the results above the MP process. The author's algorithm is identifiable as the MP, although he does not identify it as so. In a related paper [16], the same author shows that this model has benefits for speech processing. In all his research, there were no references made to the work that was done at Syracuse University work, which initially implemented the ESM via the MP. In two other papers [17, and 18], a high resolution method is used to extract the complex poles from an audio signal. The author studies the appropriate selection of the model order. He also tracks the variation of the audio signal subspace. Although the author calls this method ESPRIT, it is actually the MP algorithm.

He also neglects to reference the work performed at Syracuse University, although in [18], a reference is made to the work done in [12]. This same author in [19], foresees the problems the MP algorithm has with double poles, and attempts to build a multiple pole model for the MP algorithm to solve.

Since much research of the MP has been called other names, Table 1.1 was created to show the differences between the estimation techniques. It shows that the work done in [14, 15, 17, and 18] uses the same technique as the MP algorithm. The MP technique in [14] is called a least square problem, [15] refers to the MP as ESM, and [17 and 18] calls the MP the ESPRIT technique.

Matrix Pencil [7-13]	**Badeau [17,18] Jesper [15] Lemmerling[14]**	**Strobach [6]**	**ESPRIT [4,5]**
Direct Data	Direct Data	Direct Data	Covariance
Snapshot	Snapshot	Snapshot	Multiple Snapshot
Generalized Eigenvalue	Generalized Eigenvalue	Generalized Eigenvalue	Generalized Eigenvalue
Complex Exponential Decaying Sinusoids	Complex Exponential Decaying Sinusoids	Cisoids	Cisoids
		Array Doublets	Array Doublets

Table 1.1: The different and common techniques used in each of the estimation methods.

1.2 Speech Processing

Speech is one of the primary ways that humans communicate. Speech is produced by pushing air through ones vocal tract. For a radio transmission, the mechanical form of speech is transferred to an electrical signal and back again to a mechanical sound pressure using analog circuitry. The first type of technology to do this was the telephone. Digital technology then became the basis of modern communications. The analog to digital (A/D) converter has allowed us to digitize a speech signal. This technology makes it possible to transmit and process a digital speech signal. Digital processors with their fast speed, low cost and power, and tremendous versatility have replaced a large part of analog based technology.

1.2.1 Speech Production [20]

To build a model of the speech production mechanism, we need to first understand how speech is produced. The actual speech production system is shown in Figure 1.1. The production of speech first occurs when air pressure produced by the lungs forces air through the vocal cords that, when under tension, produce puffs of air that excite resonances in the vocal and nasal cavities. The brain and the musculature control the entire speech production process. Since speech production can be viewed as a time invariant system, it can be easily modeled. Figure 1.2 shows the model of a speech production system. A speech model consists of an excitation signal $u(t)$ that is inputted into a linear time invariant vocal tract filter $w(t)$ to produce the output speech $s(t)$. The vocal tract can be modeled as an acoustical tube. The excitation acts as a carrier signal with the vocal tract filter modulating the acoustical information onto the excitation signal.

The excitation signal convolves with the modulated vocal tract response to produce the speech signal. In the time domain this model is expressed as

$$s(t) = u(t) \otimes w(t) . \tag{1.1}$$

Transforming Eq. 1.1 to the frequency domain produces

$$\mathbf{S}(\omega) = \mathbf{U}(\omega)\mathbf{W}(\omega) . \tag{1.2}$$

Where *U(w)*, *W(w)*, and *S(w)* are the Fourier transform of the excitation signal, modulated vocal tract response, and the speech signal respectively.

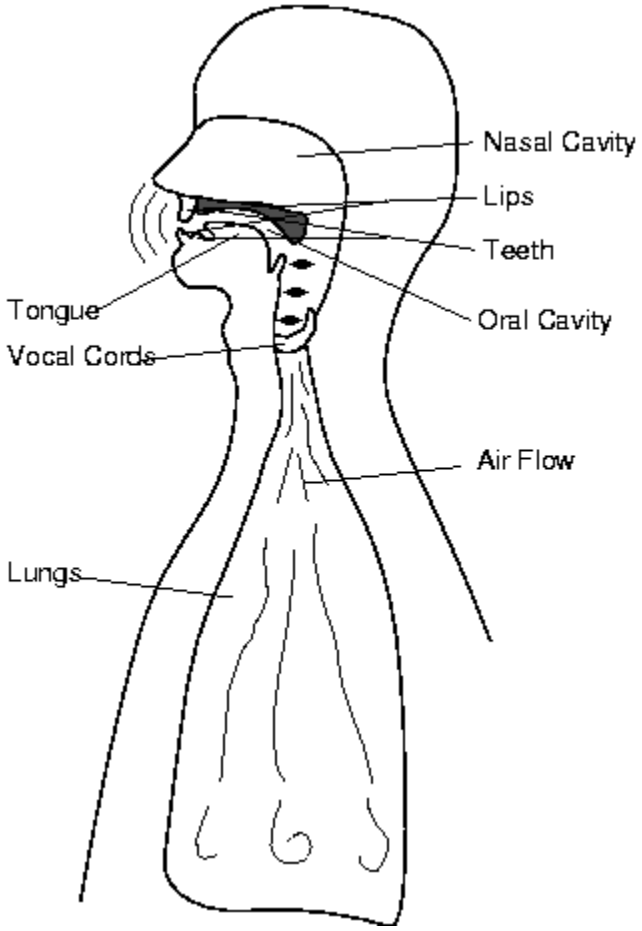

Figure 1.1: Cross section of the vocal tract.

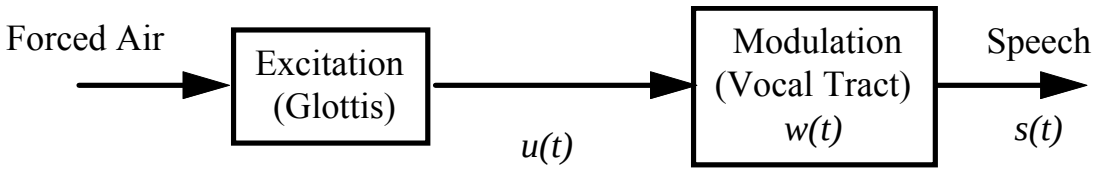

Figure 1.2: Model of the vocal tract.

There are three classes of speech that humans produce. They are voiced, unvoiced, and silence. The type of speech depends on the excitation signal. If the speech is voiced, the excitation signal consists of a train of impulses. This train of impulses are produced when forced air from the lungs travel through vibrating vocal cords. The speech is unvoiced when the excitation signal consists of white noise. This occurs when speech is produced when the vocal cord are not vibrating. Silence occurs when the excitation signal is zero.

The vocal tract is very similar to an acoustical tube. The teeth, lips, tongue, and larynx all contribute to the shape of the vocal tract. It has natural frequencies that are dependent on its shape. These natural frequencies are called formants. The formants are visible in the spectrum when voiced speech is produced.

1.2.2 Speech Modeling

There are many different speech models that are used for speech processing. The first model that will be discussed is the modulated sinusoidal model. Since speech is modeled as a slowly varying vocal tract filter with an excitation signal that is either quasi-periodic train of impulses or white noise, speech can be represented by the sum of

sinusoids. These sinusoids consist of time varying amplitude, frequencies, and phase terms. The speech signal is expressed as,

$$s[n] = \sum_{k=1}^{M} a_k[n]\cos[\theta_k[n]], \tag{1.3}$$

where the time-varying amplitude and phase terms are denoted by $a_k[n]$ and $\theta_k[n]$, respectively. The time varying frequency of each sine wave is given by the derivative of the phase, denoted by $\omega_k[n] = \theta'_k[n]$.

Another type of model that is used is an autoregressive (AR) model. In this model the signal is represented by an all pole transfer function. This transfer function represents the vocal tract transfer function. The all pole model contributes to the short time spectral envelope of the speech spectrum. In the discrete form the AR model equation is expressed as

$$\hat{s}[n] = -\sum_{i=1}^{p} a_i s[n-i]. \tag{1.4}$$

The synthesized speech $\hat{s}[n]$ is written in the form of a linear predictor of its past p weighted samples represented by $s[n-i]$. The order of the transfer function is denoted by p. The variable a_i is the set of coefficients or weights, which are solved through a set of p linear equations. Expressing Eq. 1.4 in the z-domain, the AR model is written as

$$H(z) = \frac{1}{1 + \sum_{i=1}^{p} a_i z^{-p}} \quad . \tag{1.5}$$

The all pole model is the basis of the LPC, used in many of the voice encoders (vocoders).

1.2.3 Short Time Characteristics of Speech [21]

Speech is a non-stationary process, although the properties of a speech signal will change relatively slowly with time. Given this, it is useful to separate the speech signal into short consecutive time sequences called segments or frames. These segments exhibit a quasi-stationary property, which allows one to process them accordingly. Speech segments that are too long cause the stationary property to become invalid. Segments that are too short prevent an accurate picture of the spectral features. Often, windowing is performed on the segments prior to spectral analysis. Windowing is the process of multiplying a segment of speech by a function $w[n]$ of finite duration.

Bandwidth verses leakage suppression tradeoffs exist when a speech segment is windowed. For bandwidth issues, windowing tends to broaden impulses in the theoretical Fourier representation; this causes exact frequencies to be less sharply defined. For leakage suppression issues, a windowing function should be chosen which will be compatible with our overlap and add reconstruction processing used at the output of our system. Refer to Figure 1.3 and Figure 1.4 for the time and spectral representation of the several common window functions.

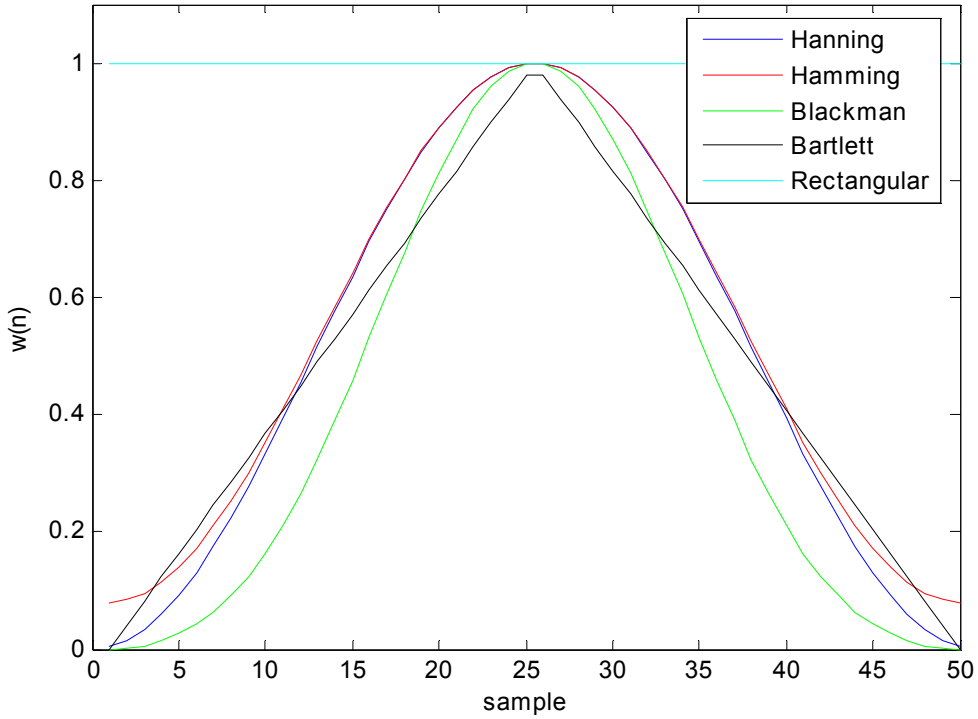

Figure 1.3: Common window functions.

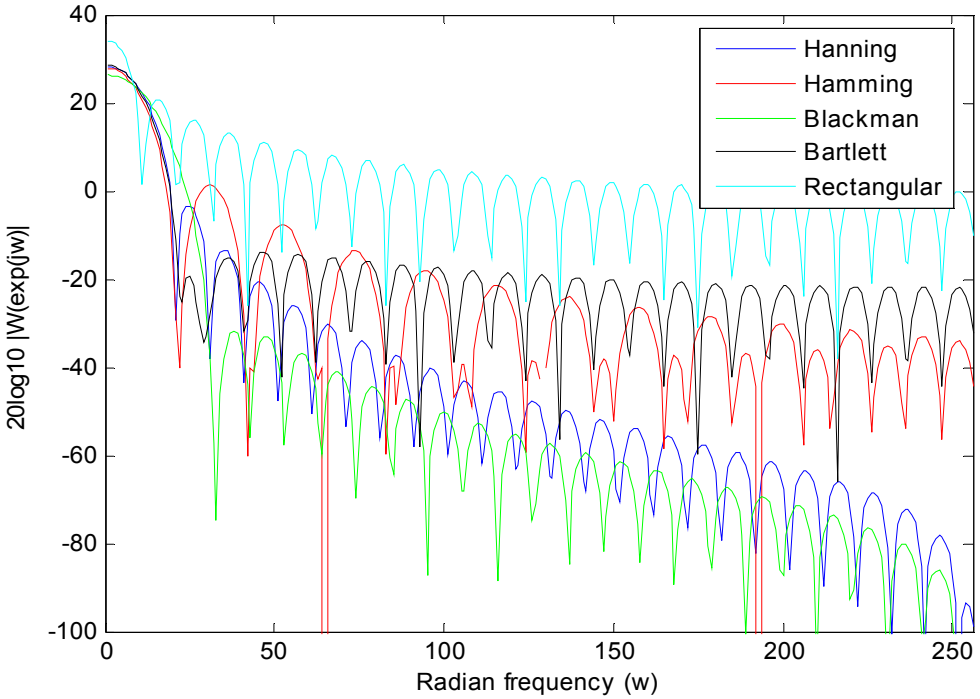

Figure 1.4: Fourier Transform (log Magnitude) of 512 sample length for the Hanning, Hamming, Blackman, Bartlett, and Rectangular window functions.

Depending on the type of speech features you need to exploit in the application, the type of windowing will affect them differently. A rectangular window is the simplest type of windowing, and is expressed as

$$w[n] = \begin{cases} 1 & 0 \le n \le N \\ 0 & otherwise \end{cases} . \tag{1.6}$$

The frame length is expressed as N, or the number of samples within that window. The main lobe of the magnitude response will decrease as the value of N increases. The rectangular window has the smallest main lobe, see Figure 1.4, of all the types of windowing that will be discussed.

The Hamming window is another type of window that is used in speech processing. The Hamming window is similar to a raised cosine pulse, and is defined as.

$$w[n] = \begin{cases} .54 - 0.46\cos(4\pi n / N) & |n| \le N/2 \\ 0 & otherwise \end{cases} . \tag{1.7}$$

The spectrum of the Hamming window can be seen in Figure 1.4. The main lobe is wider than the rectangular window's main lobe, and the side lobes have less energy. The Hamming window's disadvantage is that the side lobes do not taper to zero.

The Blackman window is another type of window that can be used in speech processing. The function which represents the Blackman window is expressed as

$$w[n] = \begin{cases} 0.42 - 0.5\cos(4\pi n / N) + 0.08\cos(8\pi n / N) & |n| \le N/2 \\ 0 & otherwise \end{cases} . \tag{1.8}$$

The disadvantage of the Blackman window is that it has a slightly larger main lobe than the Hamming. The advantage is that the side lobes taper to zero and has less energy.

A fourth type of window is the Hanning window. This type of window has a narrower main lobe when compare to the Blackman window, although it is slightly larger than the Hamming window. Unlike the Hamming window, the tails go to zero, which is a benefit when the overlap-and-add reconstruction process is applied. The Hanning window is expressed as

$$w[n] = \begin{cases} .5 - 0.5\cos(4\pi n / N) & |n| \le N/2 \\ 0 & otherwise \end{cases}. \tag{1.9}$$

The final type of window to be discussed is called the Barlett window, or the triangular window. This type of window's main lobe is similar to the Hamming and Hanning window. The side lobes have less energy than the rectangular window, but more than the other window functions. It doesn't taper to zero like the Hanning or Blackman window functions.

Of these windows, the Hamming window is the most widely used windowing function in speech processing when spectral analysis is being performed. This is due to the tradeoffs between lower frequency resolution and higher side-lobe attenuation. It will be shown that the MP algorithm is a super resolution method that avoids these window tradeoffs.

For the purpose of speech re-synthesis after windowing, the Bartlett window is frequently used. This is the case for the re-synthesis of speech when the MP algorithm is applied. With a 50 % overlap, the endpoints of the segment are weighted at zero. This process is useful in speech processing to eliminate discontinuity between frames, while properly weighting samples between end-points during re-construction (see Section

2.4.2.3). Rectangular windowing is not a good choice for the overlap-and-add reconstruction process in speech processing. Erroneous click sounds are introduced in the reconstruction of the speech.

1.2.4 Speech Processing Applications [22]

Since the beginning of the digital age, speech processing has evolved to include applications that were once unthinkable. Prior to the digital age, speech processing was used primarily for communications. Today much of the speech processing technology has grown into the areas of speech modification, speech coding, speech recognition, and speech enhancement.

Speech modification is defined as changing the speech signal to have a desired property. These modifications include changes in the pitch, the temporal characteristics, and the spectral characteristics. Applications that include temporal changes are speeding up the speech used in message playback, voicemail, and reading machines for the blind. Slowing down speech is useful for learning a foreign language. Voice Transformation is when someone's voice is altered to sound like someone else's voice. It is used to disguise a person's voice, or used in the entertainment industry. This application utilizes modifications in both the pitch and spectrum.

Speech coding is used in speech communication systems. The goal in this application is to reduce the information rate, while preserving a quality voice signal. There are three classes of speech coders. The first class is a waveform coder. This type of coder codes the waveform of the speech signal. This coder typically has a high end bit rate of 16-64 kilo-bits per second (kbps). The second class of speech coder is a vocoder. The vocoder is a model based low bit rate coder. It typically produces low quality

speech. It transmits the speech parameters of an individual's voice. The synthesized voiced is accomplished by utilizing the parameters and the particular model. The bit rate for the vocoder is in the range of 1.2-4.8 kbps. The final class is called the hybrid coder. This type of coder is partly a vocoder and a waveform coder. The hybrid coder's bit rate and quality stands in between the vocoder and the waveform coder. It usually operates at 4.8-16kbps. Speech coding is widely used in commercial, government, and military communications systems such as cell phones, satellite communications, digital television, voice over internet protocol (VoIP), voice storage, and music storage.

Enhancement is the process of improving the quality of a speech signal. There are two types of enhancement techniques that include preprocess and post-processing enhancement. Preprocessing enhancement occurs prior to the degradation of the signal, while post-processing enhancement occurs after the degradation of the signal. Preprocessing enhancement is accomplished by increasing the power of a transmitted signal; this is usually done with AM radio and TV transmission. Post-processing is the process of removing the additive noise in vehicle and aircraft communications, or digital communications; the reduction of background or speaker noise for hearing aid applications; separation of two speakers on one channel, known as a co-channel problem; or the restoration of old recordings that are degraded from age and scratches.

Recognition is defined as the exploitation of a speech signal to gain information from it. Such information may include the speaker's identity, language, dialect, or words. Speaker verification is an application to confirm a user via their voice print. This can be used to gain access, for example to his or her personnel bank account. Speaker identification is the ability to identify a person from a database of speakers. For example,

this is used to determine if a criminal is the individual on an implicative recording. Language and dialect identification is the ability to identify a speaker's language and dialect respectively. This could be used to determine where an individual is from. Word or speech recognition is the ability to recognize the words an individual is speaking. These later applications are widely used in, for example telephone call routing.

1.3 Dissertation Outline and Overview

This dissertation's goal is to research how the MP algorithm can be applied to speech processing problems. Chapter 2 will describe the mathematical theory of the MP algorithm (section 2.1). The exponentially damped or un-damped sinusoidal model (ESM) is discussed (section 2.2). With this background new research is presented regarding applications of the MP algorithm. The MP approach and how it is applied to a speech signal will be introduced (section 2.3). The problems with numerical stability and solutions to these types of problems will be presented (section 2.4). The estimation of the speech reconstruction error using the MP algorithm will be introduced (section 2.5). In chapter 3, the MP will be applied to the speech compression problem. In chapter 4, techniques used to enhance speech via the MP algorithm will be presented. A tone detection technique will be introduced (section 4.2), and several tone removal techniques will also be introduced (section 4.3). A new method called the Band Focus Matrix Pencil Temporal Reconstruction (BFMP-TR) will address the removal of unwanted tones in a speech signal (section 4.3.3). In chapter 5, a voiced/ unvoiced detection scheme is introduced (section 5.3), and a Generalized Harmonic Indicator (GHI) will be introduced (section 5.4). The GHI is an algorithm that estimates the pitch of an individual's speech.

Finally, chapter 6 will conclude with future potential speech processing work in which the MP algorithm can aide.

Chapter 2

Matrix Pencil

2.1 The Theory of the Matrix Pencil

The MP is a generalized eigenvalue problem in the form

$$\mathbf{A}\vec{x} = \lambda\mathbf{B}\vec{x} \tag{2.1}$$

or

$$(\mathbf{A} = \lambda\mathbf{B})\vec{x}\,. \tag{2.2}$$

Let **A** and **B** be two square matrices. The eigenvalue set λ(**A**,**B**), are said to be the pencil values or the roots of **A** relative to **B**. Here λ ϵ C, and is defined by λ(**A**, **B**)={z ϵ C: det(**A**-z**B**)=0}, where C is the field of all complex numbers. If λ ϵ λ(**A**,**B**), with $\mathbf{A}\vec{x} = \lambda\mathbf{B}\vec{x}$, and $\vec{x} \neq 0$ then $\vec{x}$ is referred to as an eigenvector of $(\mathbf{A} = \lambda\mathbf{B})$. For the pencil system of **A** relative to **B** to have a solution the following conditions need to exist:

1) $\vec{x} \neq 0$, that solves the equation $(\mathbf{A} = \lambda\mathbf{B})\vec{x} = 0$;

2) The $\det(\mathbf{A} = \lambda\mathbf{B}) = 0$ for all values of λ;

3) The rank of $(\mathbf{A} = \lambda\mathbf{B})$ is strictly dependent on the number of non zero values of λ.

If we now consider matrices **A** and **B** to be rectangular, *kxp*, then the $\det(\mathbf{A} = \lambda\mathbf{B}) = 0$ property is no longer a possible solution, since a determinant does not exist for a non square matrix. Therefore; the non square matrices need to be converted to square matrices. This is done by multiplying $(\mathbf{A} = \lambda\mathbf{B})$ by either $\mathbf{A}^H$ or $\mathbf{B}^H$. The superscript denotes the hermitian (conjugate transpose) operation. Once this is done we obtain

$$\mathbf{A}^H(\mathbf{A} = \lambda\mathbf{B}) = \mathbf{A}^H\mathbf{A} = \lambda\mathbf{A}^H\mathbf{B} \tag{2.3}$$

or

$$(\mathbf{A} = \lambda\mathbf{B})\mathbf{B}^H = \mathbf{A}\mathbf{B}^H = \lambda\mathbf{B}\mathbf{B}^H. \tag{2.4}$$

These two matrices are considered square, and the determinant property is possible.

To address the rank issue, let's assume again that the two matrices **A** and **B** are both of size *kxp*. The MP of this relationship would be $\{\mathbf{A}- \lambda\mathbf{B}; \lambda \epsilon C\}$. If we consider that matrix **A** and **B** to have a certain relationship, then the rank of the matrix pencil can be simplified. Let

$$\mathbf{A}= \mathbf{WXYZ} \tag{2.5}$$

and

$$\mathbf{B}=\mathbf{WXZ}, \tag{2.6}$$

where

W is a *kxm* matrix, $m \leq k$,

X is a *mxm* diagonal matrix,

Y is a *mxm* diagonal matrix, and

Z is a *mxp* matrix, $m \leq p$,

which yields

$$\mathbf{A}- \lambda\mathbf{B} = \mathbf{WXYZ}- \lambda\mathbf{WXZ} = \mathbf{WX}(\mathbf{Y}-\lambda\mathbf{I})\mathbf{Z}. \tag{2.7}$$

Therefore:

rank($\mathbf{A}- \lambda\mathbf{B}$)= rank($\mathbf{WX}(\mathbf{Y}-\lambda\mathbf{I})\mathbf{Z}$)= min{rank(**W**), rank(**X**), rank(**Z**), rank($\mathbf{Y}- \lambda\mathbf{I}$)}.

We can assume that rank(**W**)=rank(**X**)=rank(**Z**)=m, where $m \leq k$ and $m \leq p$,

and the rank($\mathbf{Y}- \lambda\mathbf{I}$)= m, provided that $\lambda_i \neq 0$ for i=1,2,…,m. If $\lambda_i = 0$ for i=m, then rank($\mathbf{Y}-\lambda\mathbf{I}$)=m-1. Therefore, the MP $\mathbf{A}- \lambda\mathbf{B}$ will also reduce in rank to m-1. By definition, λ_i's

are exactly the generalized eigenvalues of the matrix pair **A** and **B**. This definition allows us to determine the rank of **A**- λ**B**. Since both matrices span the same subspace, the generalized eigenvalues corresponding to the common null space of the two matrices will be zero. The null space exists when λ_i=0, for any i, and its corresponding eigenvectors $\vec{x}_i$ are part of this space [23, 24, 25].

Since the nonzero general eigenvalues of the matrix pair **A** and **B** are λ_i, and the corresponding generalized eigenvectors are x_i, i=1,2,…,m, Eq. 2.1 becomes

$$\mathbf{A}x_i-\lambda_i\mathbf{B}x_i=0, \text{ for } i=1,2,\ldots,m \tag{2.8}$$

or

$$\mathbf{AX}-\mathbf{\Gamma BX}=0. \tag{2.9}$$

Where $\mathbf{\Gamma}$ = diag{λ_i, i=1,2,…,m}, and **X**={ x_i, i=1,2,…,m}.

The objective is to determine the values of the generalized eigenvalues. This is done in the following manner. Let

$$\mathbf{B}^{+}\mathbf{A}=\mathbf{X}^{+}\mathbf{\Gamma}\,\mathbf{X}, \tag{2.10}$$

where $\mathbf{B}^{+}$ and $\mathbf{X}^{+}$ are the Moore Penrose pseudoinverse of **B** and **X** respectively, which is defined as $\mathbf{B}^{+}=(\mathbf{B}^{H}\mathbf{B})^{-1}\mathbf{B}^{H}$ and $\mathbf{X}^{+}=(\mathbf{X}^{H}\mathbf{X})^{-1}\mathbf{X}^{H}$.

The matrix **B** is an important matrix to solving this eigenvalue problem. First, the number of eigenvalues that exist depends on the rank of **B**. Therefore, if the rank(**B**) = k, assuming **B** is of dimension *kxp* and k < p, then there are k eigenvalues. Also, if $\lambda \neq 0$, and $\lambda \in \lambda(\mathbf{A}, \mathbf{B})$ then $(1/\lambda) \in \lambda(\mathbf{B},\mathbf{A})$. In addition to this is if **B** is nonsingular then $\lambda(\mathbf{A}, \mathbf{B}) = \lambda(\mathbf{B}^{+}\mathbf{A},I)= \lambda(\mathbf{B}^{+}\mathbf{A})$.

Otherwise, if the rank of **A** is ill conditioned, then λ(**A**, **B**) may be finite, empty, or infinite. In these cases, **B** is nearly singular, and the characteristic polynomial is of

degree less than k. The characteristic polynomial of the reciprocal problem $\mathbf{B}-(1/\lambda)\mathbf{A}$ has a zero at the origin equal in multiplicity to the defect k in the characteristic polynomial of $\mathbf{A}-\lambda\mathbf{B}$. A corresponding eigenvector for both problems is naturally a null vector of **B**. Moreover, if **B** is perturbed slightly, $\lambda(\mathbf{A},\mathbf{B})$ will generally contain k large eigenvalues that become infinite as the perturbation is reduced to zero. In this case the eigenvalues are extremely sensitive to perturbation in **B** and cannot be calculated accurately without resort to high precision arithmetic. However there may be other eigenvalues that are insensitive to perturbations in **A** and **B**.

A generalized eigenvalue can be used to solve the roots of a system. The system is dependent on the type of model used. The sum of decaying complex exponentials is one such model that can be solved by using the generalized eigenvalue. In the next section, this design is explored.

2.2 Model for the Sum of Complex Exponentials

A function such as a speech signal x(t) can be approximated as a sum of decaying complex exponentials. Herein, this signal is sampled at 8 kHz, and segmented into stationary frames. The stationary signals are used to perform short time spectral analysis. In speech, stationary frames need to be larger than the pitch period of an individual voice. Therefore, the smallest time frame is usually around 5 msec at an 8 kHz sampling rate. This corresponds to a female or child's fundamental frequency, which is approximately 400 Hz. It also needs to be less than 40 msec., because any larger than this, the characteristics of the speech will not be stationary [26].

In this work, speech will be modeled as a sum of decaying/ growing complex exponentials as in

$$x(t) = \sum_{i=1}^{M} R_i \exp(s_i t) \quad 0 \leq t \leq T. \tag{2.11}$$

The variable x(t) is the segmented speech sample, and

- R_i - are the residues or complex amplitudes
- s_i =a+bi
- $\sqrt{a^2 + b^2} = \exp(-\alpha_i)$ = damping factors
- $\tan^{-1} \frac{b}{a} = \omega_i$ = angular frequencies (ω_i =2πf)
- M = number of poles.

For the discrete case the variable t is replaced by kT_s, where T_s=1 is the normalized sampling period. This allows the discrete case to be rewritten as

$$x(k) = \sum_{i=1}^{M} R_i z_i^k \quad \text{for k=0,1,2,….,N-1 ,} \tag{2.12}$$

where

$$z_i = \exp(s_i T_s) = \exp(-\alpha_i + j\omega_i) T_s \quad \text{for i=1,2,3,….,M.} \tag{2.13}$$

These components are known as the poles.

The value R_i consists of both the gain and the phase delay of the system, and is defined as

$$R_i = A_i \exp(-j\phi_i), \tag{2.14}$$

where

- A_i = amplitude
- ϕ_i = phase delay.

Therefore, this model consists of everything needed to break up speech into its individual components, with associated amplitudes, initial phases, damping factors, and the frequencies. Further breaking down Eq. 2.12, will show these components of the model, and how they interact to make up the signal. Combing Eq. 2.13 and 2.14 into 2.12 leads to

$$x(k) = \sum_{i=1}^{M} A_i \exp(-j\phi_i) \exp(-\alpha_i + j\omega_i)^k . \tag{2.15}$$

Expanding this equation using Euler's formula leads to

$$x(k) = \sum_{i=1}^{M} A_i \exp(-k\alpha_i)[\cos(\phi_i) - j\sin(\phi_i)][\cos(\omega_i k) + j\sin(\omega_i k)] . \tag{2.16}$$

Using trigonometric product formulas and canceling like terms reduces Eq. 2.16 to an equation for real-valued signals

$$x(k) = \sum_{i=1}^{M/2} A_i \exp(-k\alpha_i) \cos(\omega_i k - \phi_i)] , \tag{2.17}$$

where M/2 is the number of sinusoids used to reconstruct the signal, and M corresponds to the number of poles. [13].

The goal is to find the best estimate of the residue R and the poles z, by using the signal x and the number of poles M, which can be varied. It will be shown how the value of M is determined. This value will determine the accuracy of the reconstructed signal, by the number of poles that are used to reconstruct x.

The MP algorithm is a super resolution spectrum analyzer. It has many benefits that the discrete Fourier transform (DFT) does not have. It should be noted that the MP algorithm is not being proposed to replace the DFT, but to help in situations where the

MP has benefits [15]. For comparison to the MP representation of Eq. 2.16, the DFT is given as

$$x(k) = \sum_{i=0}^{N-1} A(\omega_i) e^{j\omega_i k} = \sum_{i=0}^{N-1} A(\omega_i)[\cos(\omega_i k) + j\sin(\omega_i k)], \qquad (2.18)$$

where

$$\omega_i = 2\pi i/N,\ i=0,1,2,\ldots N-1.$$

It is known using the DFT, that we can get the summation of sinusoids when we have the amplitude, phase, and a presumed number of frequency bins. In the MP algorithm, we have the amplitudes, initial phases, the decay factors, and the frequencies that are determined by the data. This determination allows the frequencies to be placed exactly at the locations of the dominant frequencies. In applications, such as speech processing, there may be areas where two or three frequencies are close together. In a case similar to this, the MP algorithm will have an advantage since it will locate each frequency exactly, not within a prearranged bin as with the DFT. The DFT has lower frequency resolution that is a function of N. This is a huge difference, since the MP algorithm is adaptive to the data. The second advantage is that the MP includes a decaying factor. The MP algorithm allows one to incorporate a decaying or growing sinusoid in the signal model. The DFT does not allow one this advantage. A third advantage is the accuracy of the MP algorithm over the DFT; its variance comes closer to that of the Cramer-Rao Bound [27]. These benefits allow the MP to be described as a super resolution spectrum analyzer, and to be used to analyze data in a more adaptive manner.

2.3 MP Algorithm

There are a few different MP approaches that can be used to calculate the poles and residue of a speech signal. The first approach is discussed in the next section. This approach is called the data approach, which uses the data from the signal to generate the poles and residues. The other approach is referred to as the eigenvector approach. This method performs the SVD on the data. The resulting eigenvectors can then be used to generate the poles and residues.

2.3.1 Data Approach

Putting Eq. 2.12 in matrix form gives

$$\mathbf{X} = \mathbf{ZR}\,, \tag{2.19}$$

where a N-sample of speech is represented by

$$\mathbf{X} = \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ . \\ . \\ x(N-1) \end{bmatrix}, \tag{2.20}$$

the residue parameters for a speech segment is represented by

$$\mathbf{R} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ . \\ . \\ R_M \end{bmatrix}, \tag{2.21}$$

and the pole parameters are represented by

$$\mathbf{Z} = \begin{bmatrix} 1 & 1 & . & . & 1 \\ z_1 & z_2 & . & . & z_M \\ z_1^{\,2} & z_2^{\,2} & . & . & .z_M^{\,2} \\ . & . & . & . & . \\ z_1^{\,N-1} & z_2^{\,N-1} & . & . & z_M^{\,N-1} \end{bmatrix} NxM \quad [28] \tag{2.22}$$

Recall the MP is in the following form,

$$\mathbf{X}_2 = z\mathbf{X}_1, \tag{2.23}$$

where

$$\mathbf{X}_1 = \begin{bmatrix} x(0) & x(1) & . . & x(L-1) \\ x(1) & x(2) & . . & x(L) \\ x(2) & x(3) & . . & . \\ . & . & . . & . \\ x(N-L-1) & x(N-L) & . . & x(N-2) \end{bmatrix} (N-L)x(L), \tag{2.24}$$

$$\mathbf{X}_2 = \begin{bmatrix} x(1) & x(2) & . . & x(L.) \\ x(2) & x(3) & . . & x(L+1) \\ x(3) & x(4) & . . & . \\ . & . & . . & . \\ x(N-L) & x(N-L+1) & . . & x(N-1) \end{bmatrix} (N-L)x(L), \tag{2.25}$$

and z is the set of eigenvalues, in which for the MP analysis are the poles of the system.

The speech samples x(0) to x(N-1) are the segmented N-sample speech frame. $\mathbf{X}_1$ is a Hankel matrix from x(0) to x(N-2), and $\mathbf{X}_2$ is the same from x(1) to x(N-1). The two

matrices are related by a delay of one sample. This corresponds to a series of poles that equate the two matrices. The variable L is referred to as the pencil parameter. For speech and other applications, this value is valid within the range of M<L<N-M [28].

For estimation purposes the value for L should be within the range N/3≤L≤2N/3. This range holds for optimum estimations, dictated by the Cramer Rao Bound [29].

It can also be shown that

$$\mathbf{X_2} = \mathbf{Z_1}\mathbf{R_0}\mathbf{Z_0}\mathbf{Z_2}, \tag{2.26}$$

and

$$\mathbf{X}_1 = \mathbf{Z_1}\mathbf{R_0}\mathbf{Z_2}, \tag{2.27}$$

where

$$\mathbf{Z}_1 = \begin{bmatrix} 1 & 1 & . & . & 1 \\ z_1 & z_2 & . & . & z_M \\ z_1^{\,2} & z_2^{\,2} & . & . & .z_M^{\,2} \\ . & . & . & . & . \\ z_1^{\,N-L-1} & z_2^{\,N-L-1} & . & . & z_M^{\,N-l-1} \end{bmatrix} (N-L)xM, \tag{2.28}$$

$$\mathbf{Z_2} = \begin{bmatrix} 1 & z_1 & z_1^{\,2} & . & z_1^{\,L-1} \\ 1 & z_2 & z_2^{\,2} & . & z_2^{\,L-1} \\ 1 & z_3 & z_3^{\,2} & . & z_3^{\,L-1} \\ . & . & . & . & . \\ 1 & z_M & z_M^{\,2} & . & z_M^{\,(L-1)} \end{bmatrix} MxL, \tag{2.29}$$

$$\mathbf{Z_0} = diag\left[z_1, z_2, z_3, ..., z_M\right], \tag{2.30}$$

and

$$\mathbf{R_0} = diag\left[R_1, R_2, R_3, ...R_M\right]. \tag{2.31}$$

To show this relationship, a 3x3 matrix example will help clarify the derivation. From Eq. 2.27, the following expression is shown,

$$\mathbf{X}_1 = \mathbf{Z_1 R_0 Z_2}$$

$$\mathbf{X}_1 = \begin{bmatrix} 1 & 1 & 1 \\ z_1 & z_2 & z_3 \\ z_1^2 & z_2^2 & z_3^2 \end{bmatrix} \begin{bmatrix} R_1 & 0 & 0 \\ 0 & R_2 & 0 \\ 0 & 0 & R_3 \end{bmatrix} \begin{bmatrix} 1 & z_1 & z_1^2 \\ 1 & z_2 & z_2^2 \\ 1 & z_3 & z_3^2 \end{bmatrix} \quad \text{or}$$

$$\mathbf{X_1} = \begin{bmatrix} R_1+R_2+R_3 & R_1z_1+R_2z_2+R_3z_3 & R_1z_1^2+R_2z_2^2+R_3z_3^2 \\ R_1z_1+R_2z_2+R_3z_3 & R_1z_1^2+R_2z_2^2+R_3z_3^2 & R_1z_1^3+R_2z_2^3+R_3z_3^3 \\ R_1z_1^2+R_2z_2^2+R_3z_3^2 & R_1z_1^3+R_2z_2^3+R_3z_3^3 & R_1z_1^4+R_2z_2^4+R_3z_3^4 \end{bmatrix} = \begin{bmatrix} x(0) & x(1) & x(2) \\ x(1) & x(2) & x(3) \\ x(2) & x(3) & x(4) \end{bmatrix}.$$

Similarly, from Eq. 2.26 $\mathbf{X_2}$ is expanded to

$$\mathbf{X_2} = \begin{bmatrix} 1 & 1 & 1 \\ z_1 & z_2 & z_3 \\ z_1^2 & z_2^2 & z_3^2 \end{bmatrix} \begin{bmatrix} R_1 & 0 & 0 \\ 0 & R_2 & 0 \\ 0 & 0 & R_3 \end{bmatrix} \begin{bmatrix} z_1 & 0 & 0 \\ 0 & z_2 & 0 \\ 0 & 0 & z_3 \end{bmatrix} \begin{bmatrix} 1 & z_1 & z_1^2 \\ 1 & z_2 & z_2^2 \\ 1 & z_3 & z_3^2 \end{bmatrix} \quad \text{or}$$

$$\mathbf{X_2} = \begin{bmatrix} R_1z_1+R_2z_2+R_3z_3 & R_1z_1^2+R_2z_2^2+R_3z_3^2 & R_1z_1^3+R_2z_2^3+R_3z_3^3 \\ R_1z_1^2+R_2z_2^2+R_3z_3^2 & R_1z_1^3+R_2z_2^3+R_3z_3^3 & R_1z_1^4+R_2z_2^4+R_3z_3^4 \\ R_1z_1^3+R_2z_2^3+R_3z_3^3 & R_1z_1^4+R_2z_2^4+R_3z_3^4 & R_1z_1^5+R_2z_2^5+R_3z_3^5 \end{bmatrix} = \begin{bmatrix} x(1) & x(2) & x(3) \\ x(2) & x(3) & x(4) \\ x(3) & x(4) & x(5) \end{bmatrix}.$$

These matrices are the summation of all components up to M (in this example M=3). The summation of these components is equal to the samples of the speech signal, similar in fashion to the inverse DFT. Also this example, from Eq. 2.23, shows that

$$\mathbf{X_2} - z\mathbf{X}_1 = 0 \ , \tag{2.32}$$

therefore

$$\mathbf{Z_1 R_0}\{\mathbf{Z_0} - z\mathbf{I}\}\mathbf{Z_2} = 0, \tag{2.33}$$

where [I] is a MxM identity matrix.

This yields

$$\mathbf{Z_0} - z\mathbf{I} = 0. \tag{2.34}$$

It can be shown that if $M \leq L \leq N-M$, the rank of Eq. 2.32 will be M. However, if the Mth row of $\{[Z_0]-\lambda[I]\}$ is zero, then rank of this matrix is M-1 [28]. Hence the poles $\{z= z_i ; i=1,2,\ldots,M\}$ are the generalized eigenvalues of the matrix pencil, and therefore can be solved as a ordinary eigenvalue problem. The poles will always be complex conjugate pairs and the remaining poles will be real.

To solve the generalized eigenvalues of Eq. 2.32, the Moore-Penrose pseudo inverse (MPP) is used. The MPP is defined as

$$\mathbf{X}^{+} = \left\{\mathbf{X}^{H}\mathbf{X}\right\}^{-1}\mathbf{X}^{H}, \tag{2.35}$$

where H denotes the conjugate transpose. Therefore, the following two equations can be derived using Eq. 2.32 and 2.35. Solving for z we get

$$z\mathbf{I} = \mathbf{X_1}^{+}\mathbf{X_2}, \tag{2.36}$$

and from Eq. 2.35

$$z\mathbf{I} = \left\{\mathbf{X_1}^{H}\mathbf{X_1}\right\}^{-1}\mathbf{X_1}^{H}\mathbf{X_2}. \tag{2.37}$$

The poles z_i, in Eq. 2.12, are now solved. This leaves only the residue parameters R_i to be determined. By using Eq. 2.19 and 2.35 the amplitude can be solved directly. This equation is shown as

$$\mathbf{Z}^{+}\mathbf{X} = \mathbf{R}. \tag{2.38}$$

We have parameterized the signal x(t) according to Eq. 2.11. We now can manipulate these parameters to work to the benefit of the speech processing application we are using. These applications will be shown in the following chapters. But first, an alternative solution is presented.

2.3.2 Eigenvector Approach

In the data approach the data was used to perform the MP. In the eigenvector approach, the signal space is used. The signal space is spanned by the eigenvector of the data. These vectors and values are obtained by SVD as

$$\mathbf{U\Sigma V}^{\mathrm{H}}=\mathrm{SVD}(\mathbf{X_H}), \tag{2.39}$$

and

$$\mathbf{X_H} = \begin{bmatrix} x(0) & x(1) & . & . & x(L) \\ x(1) & x(2) & . & . & x(L+1) \\ x(2) & x(3) & . & . & x(L+2) \\ . & . & . & . & . \\ x(N-L-1) & x(N-L) & . & . & x(N-1) \end{bmatrix}_{(N-L)x(L+1)} \tag{2.40}$$

To obtain both the eigenvectors and the eigenvalues, a Hankel matrix is first formed with the segmented data, shown in Eq. 2.40. As in the data approach the variable L is referred to as the pencil parameter [28]. The matrices **U** and **V** are the left and right unitary matrices respectively. They both are composed of the eigenvectors of $\mathbf{X_H}\,\mathbf{X_H}^{\mathrm{H}}$ and $\mathbf{X_H}^{\mathrm{H}}\mathbf{X_H}$ respectively, and **Σ** is a diagonal matrix containing the singular values of $\mathbf{X_H}$. The singular values can also be represented as $\mathbf{\Sigma} = \mathbf{V}^{\mathrm{H}}\mathbf{X_H}\mathbf{U}$, where

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & \sigma_{N-L} & 0 \end{bmatrix}_{(N-L)xL} . \tag{2.41}$$

Practical considerations such as additive noise presence and computational inaccuracies result in a need to estimate the number of poles (M). The number of poles can be chosen by the user at a fixed rate, or a variable rate. The variable rate method is called the low

rank error estimate, and is discussed in section 2.5. For this time we will assume the value of M is known. This discards the small singular values, from M+1 to N-L, and their corresponding eigenvectors. These are due to noise and computational inaccuracies.

Matrices **U** and **V** represent the signal space of the segmented frame of data. Therefore; if we take the first M eigenvectors of either **U** or **V** (for this case we will use **U**) and perform the MP algorithm, the M most dominant poles of the signal will be calculated. In the data approach, the poles were not ranked in any special order. For the current approach, the SVD results in ranking the unitary matrices and hence the poles. The algorithm would be performed as follows:

$$\mathbf{U} = \left[\vec{u}_1, \vec{u}_2, \vec{u}_3, ..., \vec{u}_M\right]^T \tag{2.42}$$

$$\left\{\mathbf{U_2} = z\mathbf{U_1}\right\}_{LxM} \Rightarrow \left\{\mathbf{U_1}^{+}\mathbf{U_2} = z\mathbf{I}\right\}_{MxM}, \tag{2.43}$$

where

$$\mathbf{U_1} = \left[\vec{u}_1, \vec{u}_2, \vec{u}_3, ..., \vec{u}_{M-1}\right]^T \tag{2.44}$$

is **U** with the last column deleted,

and

$$\mathbf{U_2} = \left[\vec{u}_2, \vec{u}_3, \vec{u}_4, ..., \vec{u}_M\right]^T; \tag{2.45}$$

is **U** with the first column deleted, therefore

$$z\mathbf{I} = \left\{\mathbf{U_1}^H\mathbf{U_1}\right\}^{-1}\mathbf{U_1}^H\mathbf{U_2}\ . \tag{2.46}$$

The variable z, in Eq. 2.46, represents the individual eigenvalues which are the poles of the signal. The rank of this equation is M, only if all the values of z are non zero.

Similar to the data approach, once the poles are known, the residues can be solved by using Eq. 2.19 and 2.35. The equation would be used to finally calculate the residues, similar to Eq. 2.38. This allows the signal to be represented by both the residue (amplitude and initial phase) and the poles (frequency and decay). By using the most dominant poles of the signal, the signal can be reconstructed using fewer parameters and at the same time leaving a minimum error. This error will depend on the poles that were the least dominant. The number of poles used will determine the extent of the error; this will be discussed in the next section. Due to the control over the selection of M, the eigenvector approach has many benefits in speech compression, pitch estimation and noise removal. These types of applications will be looked at in the subsequent chapters and therefore this SVD based approach is used.

2.4 MP Computational Stability with Speech Processing

A few problems occur when processing speech with the MP algorithm. The first is a floating point problem, associated with the frame size. Another problem is dealing with ill-conditioned matrices. These problems are addressed in this section.

2.4.1 Numerical Considerations

When calculating the residues of the signal with a high number of singular values, it was found, at times, that the numbers exceed the largest positive floating point number in Matlab (1.7977e+308). Also, a similar situation would occur if the numbers were

smaller than the smallest floating point number (2.2251e-308). This occurs when determining the inverse of **Z** (pole matrix) via the Moore Penrose method in Eq. 2.38.

One solution to this problem is to use a smaller matrix size (smaller window). This would lower the power the pole is raised to. This will ensure that the values inside the **Z** matrix would fall within the floating point range. Once this is achieved, taking the inverse of this matrix is not a problem, if it is within rank. The rank issue is addressed later in this section.

Frame sizing avoids other problems when combining speech processing with the MP algorithm. The smaller the frame size the quicker it is to compute the eigenvalues, eigenvectors, and residues. This is shown in Table 2.1. In this table one can see the time it takes to process a speech file at different frame sizes. The processing was performed on a Pentium 4, 1.80 GHz computer using Matlab. The larger the frame size, the more eigenvalues and eigenvectors computed. The computation of these parameters is extensive, once the frame size is greater than or equal to 20 msec.

Comparison of different frame size vs. processing time for a similar number of poles (speech file =26000 samples @ 8kHz)				
Number of poles processed per speech file	2.5 ms window (sec)	5 ms window (sec)	10 ms window (sec)	20 ms window (sec)
11,000	50.5	95.3	264.8	957.5
9500	50.0	75.0	190.2	729.6
7800	45.6	58.1	136.6	462.3
6000	36.3	42.1	88.7	282.0
5000	30.5	32.3	66.9	207.3

Table 2.1: Processing times for different frame sizes.

Another reason the frame size needs to be considered is due to the pitch period. In a voiced frame, the frame size needs to be large enough to cover the length of a pitch period of a speaker. Typically a female speaker, similar to a child's voice, has the

shortest pitch period. On average this type of speaker would have approximately a 400 Hz pitch (2.5 msec). A male speaker averages between 200 Hz to 300 Hz (3.3msec to 5msec). Thus the minimum size should not be lower than 5 msec. This would ensure a full pitch period would be within a speech frame. Therefore, the range of the frame size needs to accommodate these factors. For a best fit frame size, it should be larger than 5 msec, and smaller than 20msec, or so, when sampling at 8kHz.

2.4.2 Condition, Stability, and Location of the Poles

The condition of the poles is critical to the MP solution. By definition an ill-conditioned matrix occurs when the ratio of the largest to the smallest singular value (condition number) is too large. Large condition numbers indicate a nearly singular matrix. The condition number is a measure of stability or sensitivity of a matrix (or the linear system it represents) to numerical operations. The results of computations on an ill-conditioned matrix should not be trusted, since there maybe more than one solution to the matrix inversion. Matrices with condition numbers near 1 are said to be well-conditioned. Matrices with condition numbers much greater than one (1/eps), eps being the floating point relative accuracy, are said to be ill-conditioned [30].

The inverse of matrix cannot be determined if it is not well conditioned. Small changes in the problem data will induce relatively large changes in the solution. The MP algorithm relies on determining the inverse of a matrix when solving for the poles and the residues in Eq. 2.19 and 2.43 respectively. It has been found that Eq. 2.43 (when determining the poles) is always well conditioned, since the eigenvectors are orthogonal.

When the column space or row space of a matrix is orthogonal, then the matrix is well conditioned. The problem usually occurs when solving for the residue. In this case the Z matrix multiplied by its transpose, Eq. 2.46 needs to be well conditioned in order to solve for the residues.

The poles are complex, and consist of the frequency and decay. The amplitude of the pole is typically a value of one (no decay), and is on the unit circle. When the pole is far outside the unit circle the signal becomes unstable. This implies that the speech frame will have high growth within the frame. This occasionally occurs during a transition from an unvoiced segment to a voiced segment or during similar impulsive events. How far the pole is outside the unit circle will indicate if it is unstable. This increases the condition number of the matrix, since the pole will be raised to the N-1 power. In addition, when it is multiplied by its hermitian, the power is raised to 2N-2, which is very large, and in some cases larger than the maximum real number (1.7E+308). Therefore, a slight increase in such poles, outside the unit circle, will rapidly elevate the condition of the matrix.

A problem which is related to the above situation involves the upper or lower quantization bound of an audio signal. Unlike most audio processing algorithms, the MP algorithm uses the decaying factor as a parameter. This presents a problem when an audio frame has a pole that grows fast and exceeds the quantization ceiling. Given finite precision there is a value for the largest growth that can be handled. Since audio is 16 bit, the largest quantization value possible, without clipping is $2^{16} / 2 = 32767$. The value for the largest growing factor can be calculated, and is dependent on the frame size. It is

known that the growing factor is expressed as $\exp(-\alpha_i n)$, with n being the sample of the frame. Therefore from an initial amplitude of 1, if $\exp(-\alpha n) = 32767$, the ceiling of a 16 bit audio signal, the calculated value for the decaying factor is $\alpha = -10.39752 / N$, where N is the size of the frame. Table 2.2 shows the maximum growth with respect to the frame size. This is additional reason that the frame of a signal needs to be low. If the frame is increased, the growth factor needs to be decreased to prevent signal over-shoot. This would particularly be an issue in a fixed-point processor implementation. However, it does give us a guide for the Matlab implementations used herein. Note that in Matlab, when quantized wave files are read, the resulting signal is normalized to a maximum amplitude of ±1.

Frame Time (sec)	**Frame Size (samples)**	**α=- log(32762)/N**	**exp(-α)**
0.0025	20	-0.519859	1.68179
0.005	40	-0.259929	1.296839
0.01	80	-0.129965	1.138788
0.02	160	-0.064982	1.06714
0.025	200	-0.051986	1.053361
0.03	240	-0.043322	1.044274

Table 2.2: Comparison of frame size to maximum pole radius.

2.4.2.1 Ill-Condition Examples and Observations

The following is an example when the matrix is ill-conditioned. This example is a 160 sample frame (20 msec.). After decomposition, reconstruction as in Eq. 2.12 can be performed for error analysis. For this speech frame, the MP process does not always reconstruct correctly. Figure 2.1 shows the original and reconstructed signal for this

frame. One can see the impulse at the end of the frame; this is a transition from an unvoiced to a voiced segment. With a pencil parameter of 80, 79 poles is the maximum number of poles that can be used, the condition of the matrix gets large with any number of poles greater than 6. Referring to the z-plane plot, Figure 2.2, the pole outside the unit circle has a magnitude of 1.2216, which exceeds the limit of 1.06714. This pole causes the condition of the matrix to increase. This dominant and unstable pole causes an ill-condition situation for any number of poles greater than 6. As the number of poles increases, some of the results may be inaccurate depending on the number of poles one chooses, as can be seen in Figure 2.3. In Figure 2.3, with the number of poles

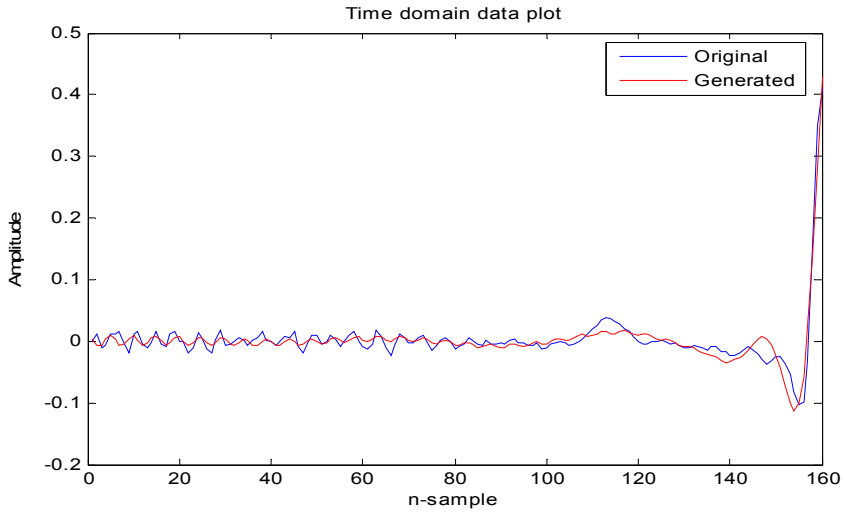

Figure 2.1: A 20 ms frame, using 6 poles for signal reconstruction out of a possible 79 poles.

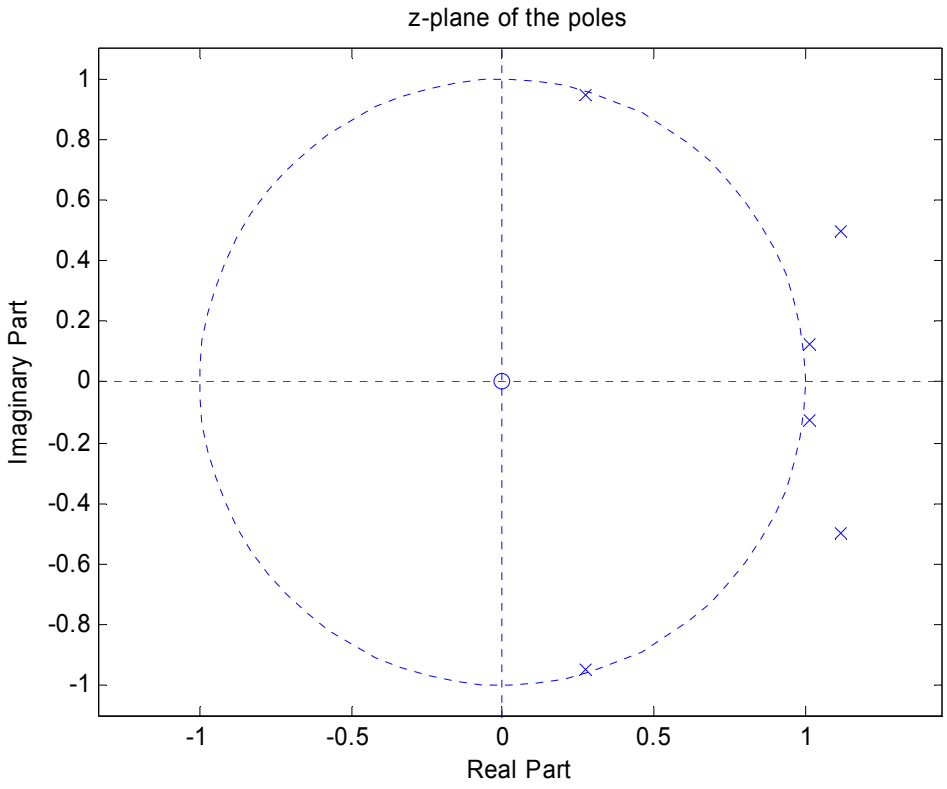

Figure 2.2: A z-plane plot of the poles for the corresponding plot of Figure 2.1.

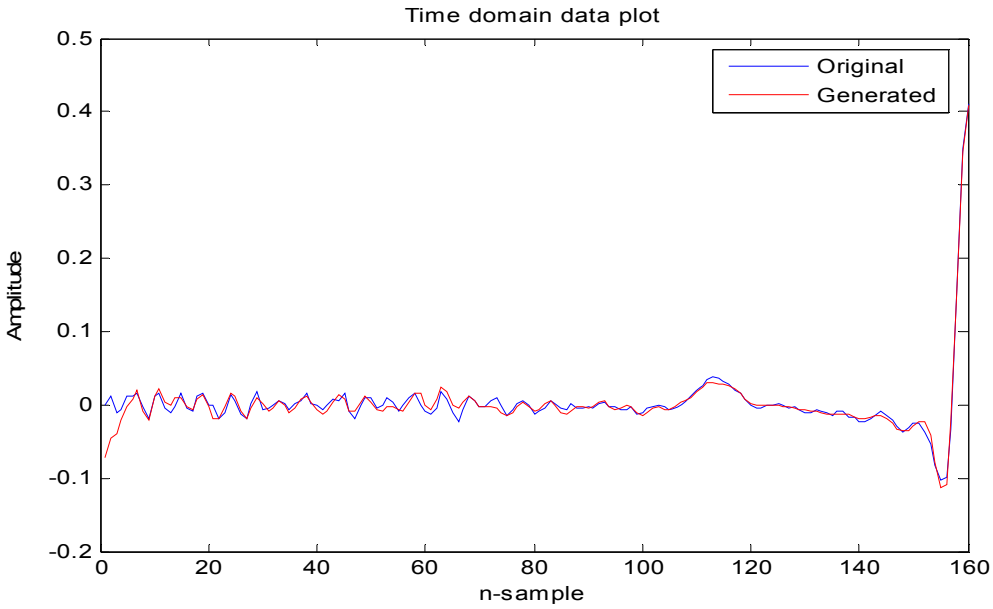

Figure 2.3: A 20 ms frame, using 27 poles for signal reconstruction out of a possible 79 poles (Ill-conditioned).

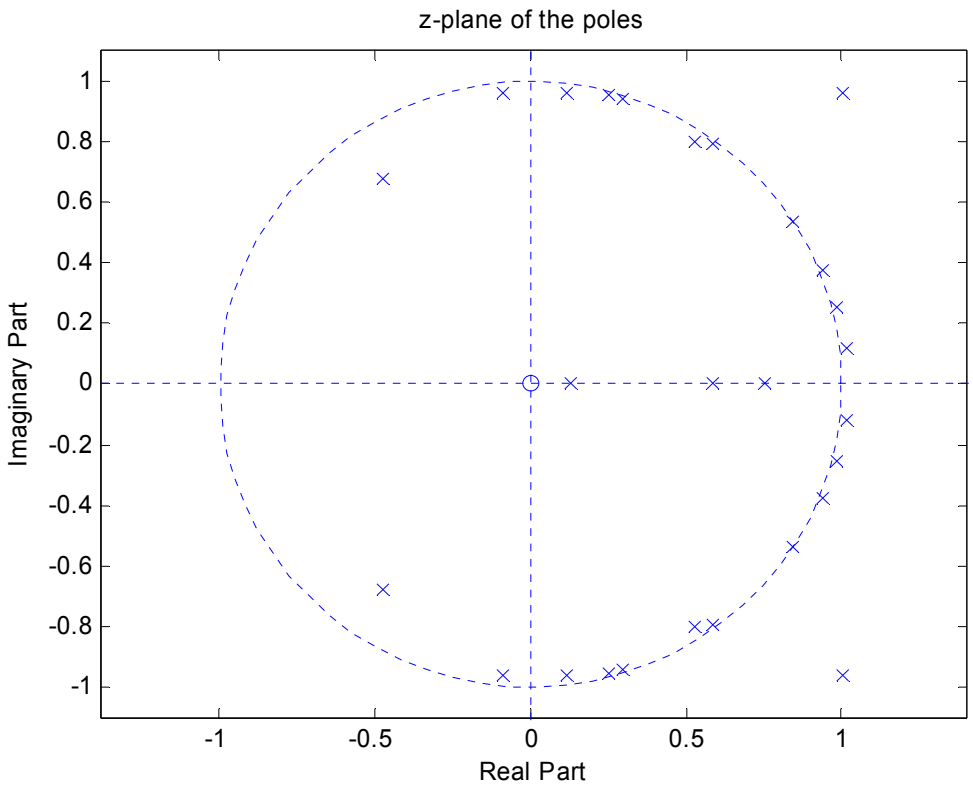

Figure 2.4: A z-plane plot of the poles for the corresponding plot of Figure 2.3.

is equal to 27, the results are not accurate. Observe that in this example, the beginning edge is inaccurate. The inner poles on the real axis together with the unstable poles outside the unit circle contribute to this error. When the numbers of poles are increased to 28, the result is fine. This is seen in Figures 2.5 and 2.6. The poles on the real axis get paired with another pole as a complex conjugate, eliminating the error. As the number of poles is increased to 29, the poles on the real axis reappear, making the results once again flawed. Out of all the number of pole possibilities, up to a total of 79, these two are the only ones that give inaccurate results. The common characteristic of the two inaccurate examples is that the location of a pole is far outside the unit circle, and there are poles on the real axis near zero, to be consistent with Eq. 2.17 for real-valued signals. In our

example, the number of poles is equal to 27 and 29 (Figure 2.3, 2.4, 2.7, and 2.8), and this characteristic occurs making the matrix ill-conditioned. Whereas, when the number of poles equal 28, Figure 2.5 and 2.6, there are no poles on the real axis near zero, and the matrix is marginal well conditioned, thus the signal is reconstructed accurately. It has also been observed that this characteristic can occur when the number of poles is even. In this case two poles are on the real axis, while the others are complex conjugate of each other, at least one being unstable. In Figure 2.9, the mean square error (MSE) between the original and reconstructed signals is presented for the number of poles retained. This plot shows the MSE jumps for a specific number of poles. An inverse linear progression would be expected, as the number of poles increases. The ill-conditioned system doesn't always give bad results, as in this case. Two of the 79 possibilities were inaccurate. Solutions to this problem are presented in the next two sections. One is using the backward method

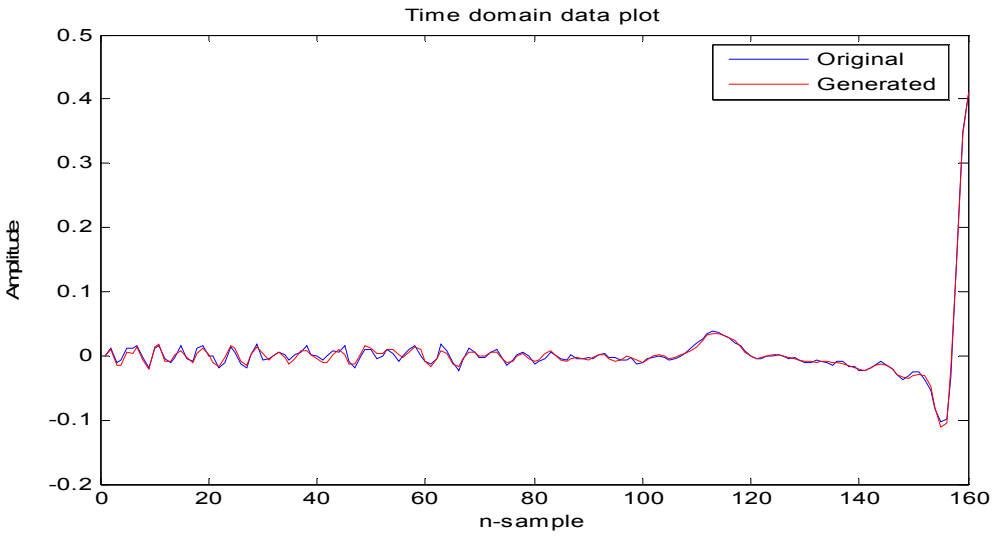

Figure 2.5: A 20 ms frame, using 28 poles for signal reconstruction out of a possible 79 poles.

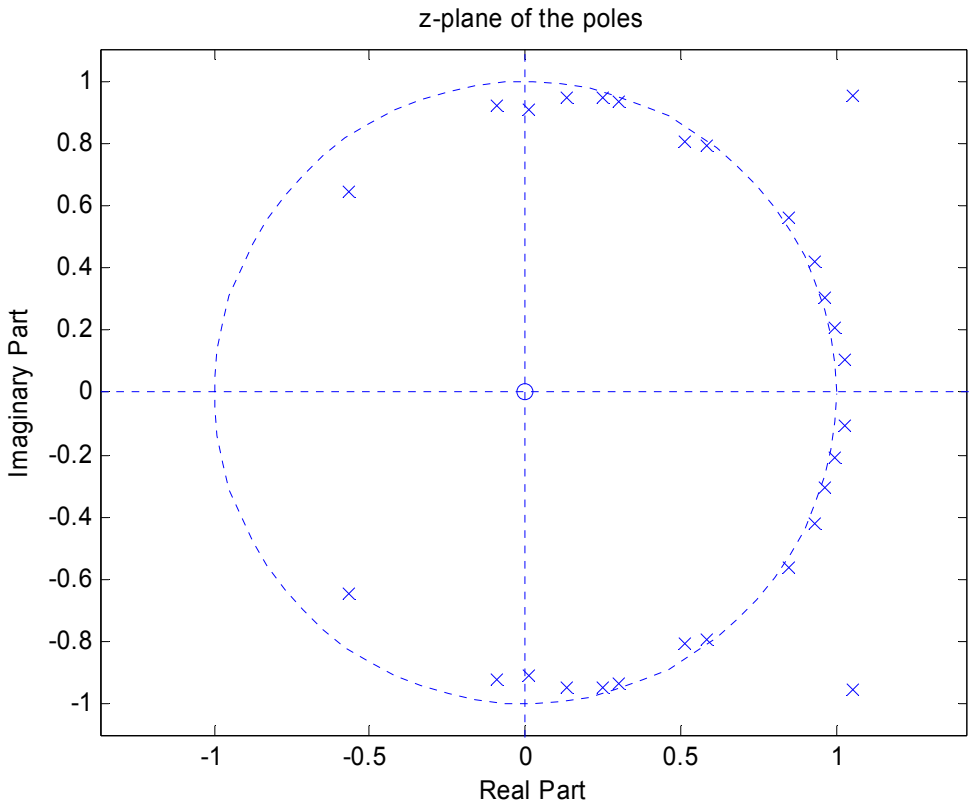

Figure 2.6: A z-plane plot of the poles for the corresponding plot of Figure 2.5.

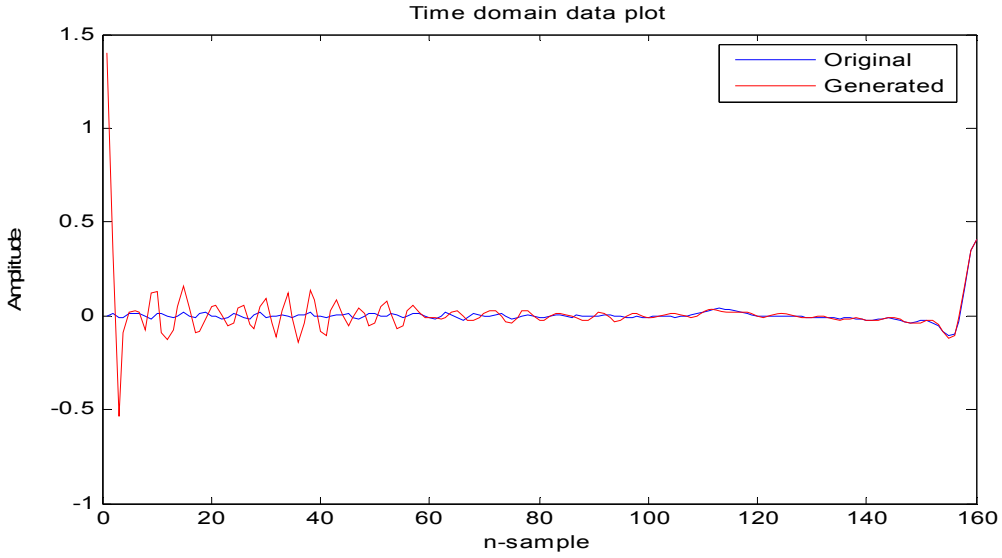

Figure 2.7: A 20 ms frame, using 29 poles for signal reconstruction out of a possible 79 poles (Ill-conditioned).

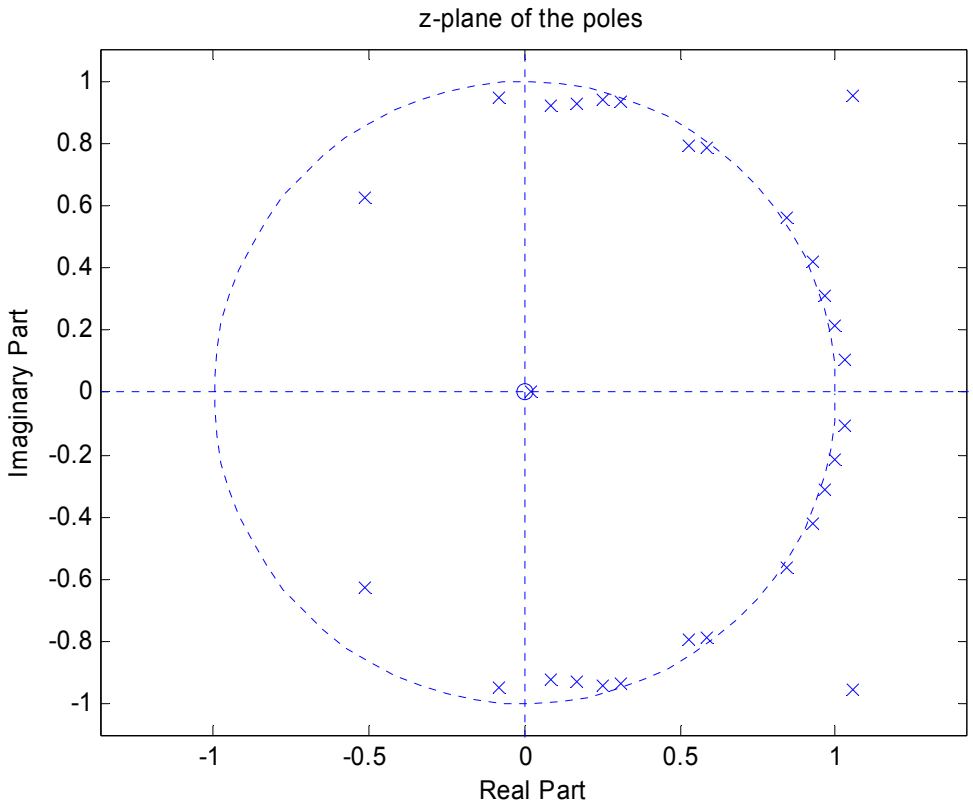

Figure 2.8: A z-plane plot of the poles for the corresponding plot of Figure 2.7.

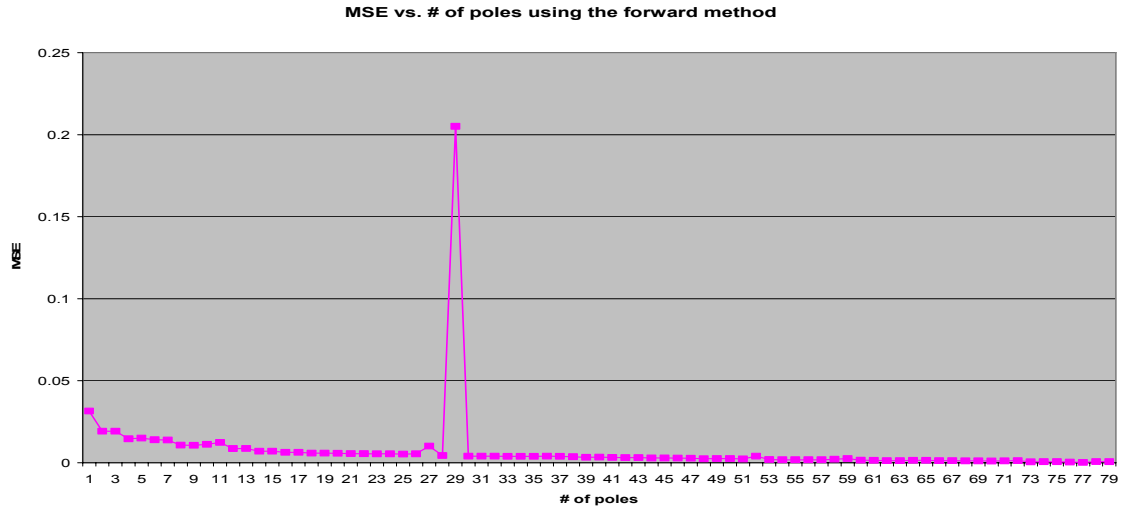

Figure 2.9: The mean square error between the reconstructed and the original signal, for a frame of data reconstructed at a different number of poles.

to avoid the rapid growth in the forward direction. The other technique which helps mitigate ill-conditions and more generally keeps reconstruction smooth at frame boundaries involves the use of a tapered window and frame overlapping [31].

2.4.2.2 Ill-Conditioning Handling: Backward Method

One solution to the ill-conditioned problem is to use what will be referred to as the backward method [32]. The backward method simply reverses the input data, and processes it in that direction. The intent is to avoid fast growths by turning these into a rapid decay. This works for speech because the chosen frame size is small enough not to include both decay and growth. So if the frame is 160 samples long, sample 160 will be sample 1 and sample 1 becomes sample 160. Once the frame is reversed the MP algorithm is applied to the backward signal. Once the backward signal is reconstructed, it is reversed once again to obtain the forward reconstructed signal. This process adds very little processing time to the overall algorithm, and is very effective.

Using the example of the previous section, Figure 2.10 shows the original signal, and reconstructed signal using the backward method when the number of poles is 29. As one can see the ill conditioned error no longer exists. The z-plane plot, in Figure 2.11, shows the pole that was outside the unit circle is now inside. The backward direction allows the pole to decay instead of grow thus keeping the condition of the matrix low. Therefore when a pole is found far outside the unit circle, one can employ the backward method to this frame.

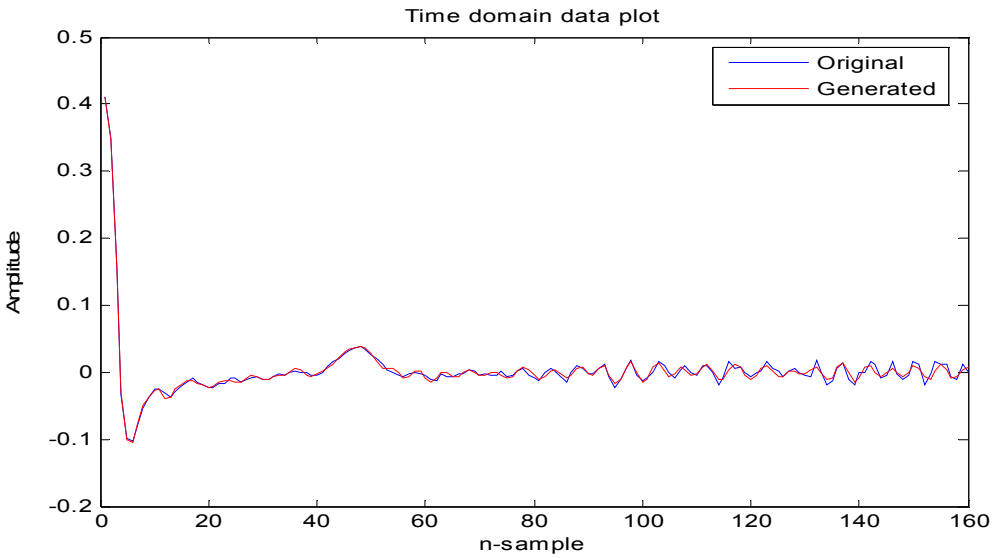

Figure 2.10: A 20 ms frame, in backward mode, using 29 poles for signal reconstruction out of a possible 79 poles.

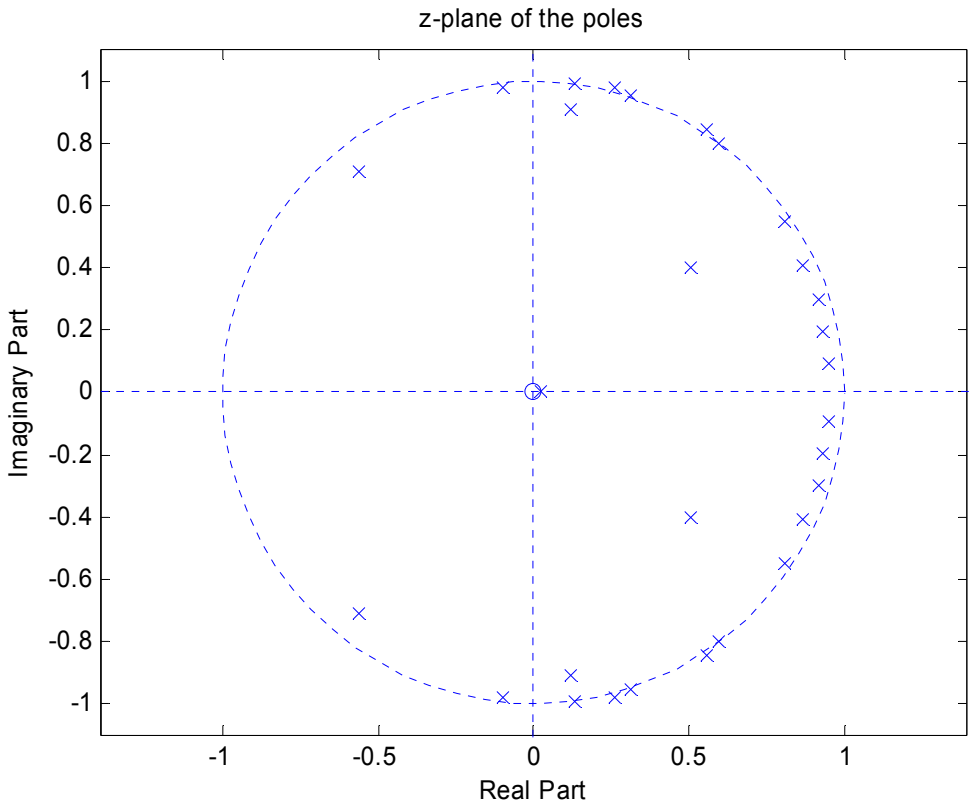

Figure 2.11: A z-plane plot of the poles for the corresponding plot of Figure 2.10.

Another potential situation is when the signal decays fast in the forward direction, this would allow the signal to grow fast in the backward direction, causing an ill-conditioned problem in the backward method if used. This problem has not appeared in this research, since speech is observed to grow faster than it decays. This would appear within reason, since attenuation is dependent on the environment, and not on the type of speech and the speaker, as would be the case for growth.

An alternative solution to the ill-conditioned problem was tried with moderately unsatisfactory results. This was to leave the growing pole out of the calculation. This solved the ill-conditioned problem since the high growth pole did not contribute to the calculation. The problem with this solution was that the MSE of the reconstructed signal increased, when comparing it to leaving the pole, risking inaccurate results. The comparison is shown in Figure 2.12 and 2.13. In Figure 2.12, the frame had a dominant pole outside the maximum radius. The MSE is compared with the different methods that were discussed. The backward method was the most favorable method, since the MSE was low and consistent. When the pole was exempted in the calculation, the MSE is increased. This is due to the removal of a strong sinusoid that makes a large contribution to the signal. In Figure 2.13 a different frame was analyzed. A less dominant pole, ranked as the 61^{st} pole, was exempt. In this case, the effect does not show any difference until the pole plays a role in the calculation. Also, the forward method results were good. Therefore in such a scenario it is best to use the backward method to reassure the results in the forward method.

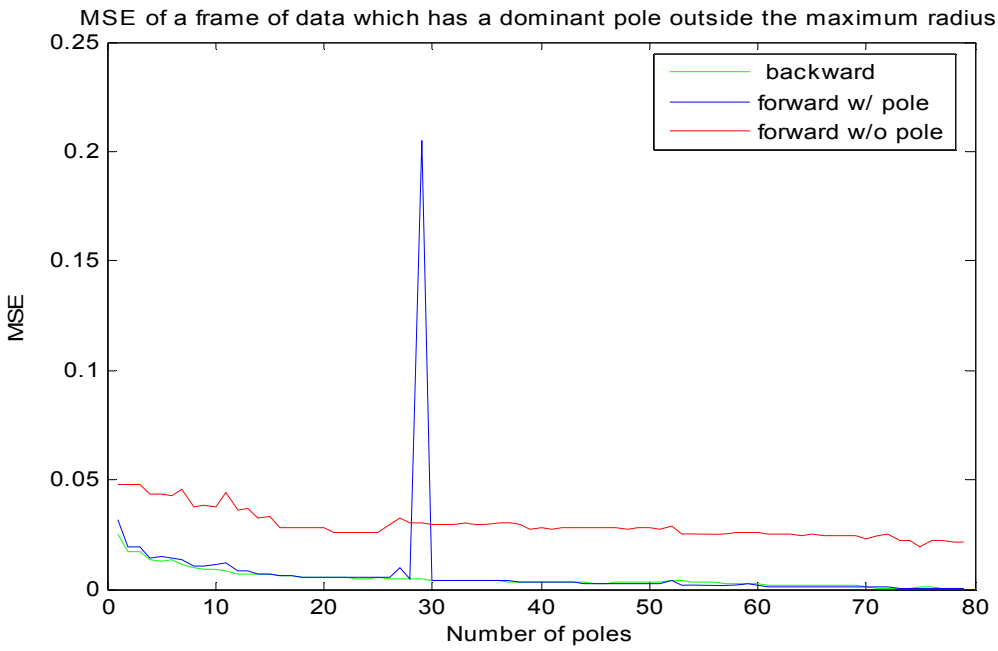

Figure 2.12: The MSE of a reconstructed frame of audio using 3 methods. The frame of data had a dominant pole outside the maximum radius.

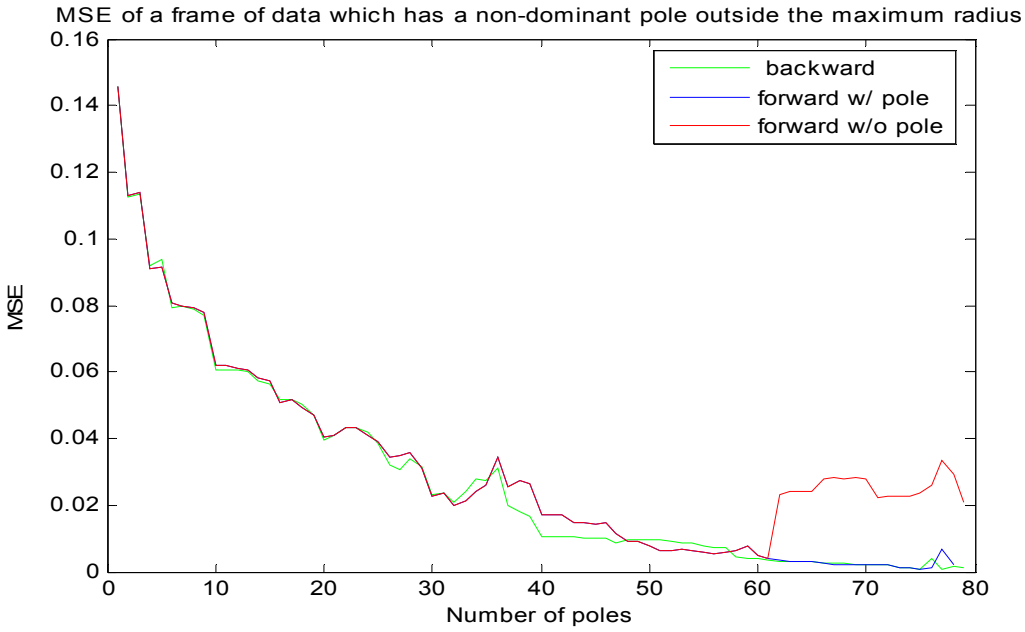

Figure 2.13: The MSE of a reconstructed frame of audio using 3 methods. The frame of data had a non-dominant pole outside the maximum radius. Note: The blue curve follows the red curve up to 61 poles.

2.4.2.3 Ill-Condition Handling: Triangular Window Weighted Algorithm

The last method which helps alleviate the ill-conditioned problem is to use a weighted triangular window method with a 50% overlap and add. As explained, we know that the ill-conditioned frames have reconstruction problems at the edges. This is due to the high growth or large decay of a particular sinusoid. To overcome this problem, a weighted triangular filter is used on each frame, with 50% overlap and add. This weighted window smoothes the edges of each frame and is expressed as[33].

$$w[n] = \begin{cases} \dfrac{n}{N/2} & \text{n = 0,1,.....,N/2} \\ w(N-n) & \text{n = N/2,.....,N - 1} \end{cases} \tag{2.47}$$

The smoothed signal has better reconstructed audio without any clicks that were present prior to the smoothing algorithm. This triangular window is shown in Figure 2.14, with each window frame color coded, and in this case three frames are represented. This technique is performed at the expense of coding efficiency due to the overlap. However, the clarity and smoothness of the reconstructed speech signal are useful for many other speech applications. For low bit rate audio coding to be effective, the frame size would need to increase more than the recommended sizes that were presented. Another use is when some poles are exempted prior to reconstruction. This will be seen to be very useful for tone and distortion removal. Therefore Triangular weighting is often required for reconstruction in general. Also, to achieve higher compression rates, one could consider trapezoidal weighting to allow for less than 50% overlap.

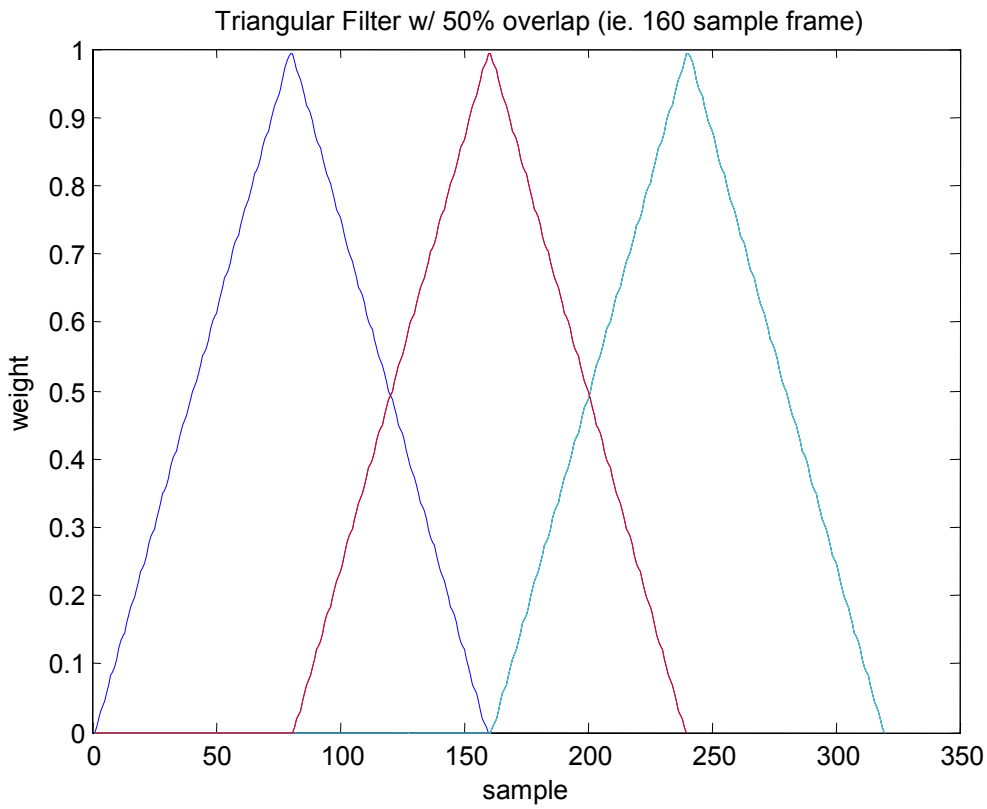

Figure 2.14: Triangular Filtering with 50% overlap. Edges of the frame have less weight than the middle of the frame.

2.4.3 Double Poles and Close Proximity Singular Values

The MP algorithm is based on a single pole model. For each frequency represented in the signal, a single pole represents a single frequency. A double pole is defined when two poles represent a single frequency. Double poles occur in a more general class of signals, involving both amplitude and frequency modulation. A physical example of a double pole model is a critically damped simple harmonic motion of a spring/mass system. In audio, a double pole would occur during a rapid growth and a rapid decay for a specific frequency within a short time window. An example would be

violin vibratos or a guitar attack [19]. In this dissertation, the emphasis is on speech processing, and not audio processing (mixtures of speech and music signals). Rarely does a double pole occur for a speech signal, particularly when frame sizes are small. But if it does occur, it may affect the reconstructed signal, producing a large error.

Signal segments with such components can present processing problems using the MP model. A heuristically found solution to this problem is to detect if there are singular values in close proximity to each other. Observe that each singular values results in a pole. This allowed for a check to determine if the number of estimated poles should stay $M=\hat{M}$ or be increased to $M=\hat{M}+1$. In this research it has been found that a 1% difference or smaller between the two singular values would determine if the number of poles should be $M=\hat{M}+1$, otherwise it is kept at $M=\hat{M}$. Theoretical investigations into this phenomenon are reserved for future research.

2.5 Reconstruction Error

Whenever a signal is represented by fewer parameters than the original, an error in the signal is encountered. This error depends on the energy of the signal as well as the number of parameters that are used to represent the reconstructed signal. In the following subsections, a novel approach is discussed to estimate the number of MP parameters (M) needed in the reconstruction of a speech signal, given a relative error.

2.5.1 Relative Error

The relative error can be determined in two ways. The first way is very computationally time consuming, and is the typical way of finding the error. Using Parseval's theorem, the error is defined as

$$error = \sum_{n=1}^{N} (x(n) - \hat{x}(n)), \tag{2.48}$$

where N is the length of the frame, and the relative error is defined as

$$relative\ error = \frac{(error)^2}{\sum_{n=1}^{N} x(n)^2}. \tag{2.49}$$

The relative frame error is determined first by using 1 pole and then calculating its reconstructed error with respect to the original signal. This is done for each increment number of poles until the error is satisfied. Recall that the obtained singular values are in decreasing order. It is time consuming since the selections of poles are used to make up the matrix Z, then the amplitudes are calculated, and finally the relative error is determined. This is done many times for each frame. The bigger the frame the more poles are used, which increases the size of the matrix Z. Also it increases the number of increments needed to satisfy the relative error.

A second and more efficient method in determining the relative error, under the matrix pencil, is to determine the number of singular values before calculating the poles. This method is called low rank modeling. The singular values, in Eq. 2.41, determine the number of poles used in the calculation again. Once the singular values are calculated, they are ranked from largest to smallest. The sum of the squared singular values, which

are not used, gives the approximation of the error. This is seen in the following equations.

Let the original signal, in hankel matrix form from Eq. 2.39, be represented by **Y**. The reconstructed signal is represented by $\hat{\mathbf{Y}}$ as

$$\hat{\mathbf{Y}} = \mathbf{U}\hat{\mathbf{\Sigma}}\mathbf{V}^{H}, \tag{2.50}$$

where

$$\hat{\mathbf{\Sigma}} = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_M & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} (N-L)xL. \tag{2.51}$$

The reconstructed signal is represented by M parameters, while the original signal is represented by using all of the eigenvectors and singular values (N-L). Therefore the error is represented by

$$\mathbf{Error} = \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^{H} - \mathbf{U}\hat{\mathbf{\Sigma}}\mathbf{V}^{H} \quad [34]. \tag{2.52}$$

This leads to

$$\mathbf{Error} = \mathbf{U}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{M+1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_P \end{bmatrix}\mathbf{V}^{H} \tag{2.53}$$

or

$$\mathbf{Error} = \mathbf{U}\hat{\mathbf{\Sigma}}_{\mathbf{P-M}}\mathbf{V}^H, \tag{2.54}$$

where P=N-L, and

$$\hat{\mathbf{\Sigma}}_{\mathbf{P-M}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{M+1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_P \end{bmatrix} . \tag{2.55}$$

If the error is multiplied by its hermitian then

$$\mathbf{Error} * \mathbf{Error}^H = \mathbf{U}\hat{\mathbf{\Sigma}}_{\mathbf{P-M}}\mathbf{V}^H * \left[\mathbf{U}\hat{\mathbf{\Sigma}}_{\mathbf{P-M}}\mathbf{V}^H\right]^H, \tag{2.56}$$

in which it simplifies to

$$\mathbf{Error} * \mathbf{Error}^H = \mathbf{U}\hat{\mathbf{\Sigma}}_{\mathbf{P-M}}\mathbf{V}^H\mathbf{V}\mathbf{\Sigma}_{\mathbf{P-M}}{}^H\mathbf{U}^H . \tag{2.57}$$

In Eq. 2.57 the unitary matrix **V** is an orthonormal matrix ($V^HV=I$), where I is an identity matrix.

Therefore Eq. 2.57 reduces to

$$\mathbf{Error} * \mathbf{Error}^H = \mathbf{U}\hat{\mathbf{\Sigma}}_{\mathbf{P-M}}\hat{\mathbf{\Sigma}}_{\mathbf{P-M}}\mathbf{U}^H . \tag{2.58}$$

The expected value of the results of Eq. 2.58 is

$$E\left[\mathbf{Error}^H * \mathbf{Error}\right] = E\left[\hat{\mathbf{\Sigma}}_{\mathbf{P-M}}\mathbf{U}^H\mathbf{U}\hat{\mathbf{\Sigma}}_{\mathbf{P-M}}\right] \tag{2.59}$$

or

$$E\left[\mathbf{Error}^H * \mathbf{Error}\right] = E\left[\sum_{i=M+1}^{P}\sum_{j=M+1}^{P}\sigma_i\sigma_j u_i^H u_j\right]. \tag{2.60}$$

The symbols σ are the individual eigenvalues which lie on the diagonal of Σ, and u are the individual eigenvectors from the matrix U. This equation can be further reduced as

$$E\left[\mathbf{Error}^H * \mathbf{Error}\right] = \sum_{i=M+1}^{P} \sum_{j=M+1}^{P} E[\sigma_i \sigma_j u_i^H u_j] \ . \tag{2.61}$$

Since the eigenvectors u_i and u_j are orthonormal, where

$u_i^H * u_j = 1$ when i=j

and

$u_i^H * u_j = 0$ when i≠j

therefore $U^H U = I$, giving

$$E\left[\mathbf{Error}^H * \mathbf{Error}\right] = \sum_{i=M+1}^{P} \sigma_i^2 \ . [35] \tag{2.62}$$

Using the inner product property, the error squared can be reduced to the norm squared of the vectors in the error matrix, as [36]

$$E\left[\mathbf{Error}^H * \mathbf{Error}\right] = E\left[\| \mathbf{Error} \|^2\right] = \sum_{i=M+1}^{P} \sigma_i^2 \ . \tag{2.61}$$

Therefore, by squaring the unused singular values of the Hankel matrix and summing them, a good approximation of the error can be determined. The associated error matrix is in Hankel form. A single number, the true error, can be computed by taking the normalization of each row, squaring them and finally taking the average value. The summation of the unused squared singular values approximates this true error.

This method, referred to herein as low rank error modeling, is a quick and feasible way to calculate the error prior to reconstructing the signal. There is no wasted time in processing pole matrices and amplitude vectors. One may determine how many poles are needed in a frame of speech, within a certain error criteria, prior to any complicated

processing. This allows the user to determine how many poles are needed for a single frame of data, and is in this sense adaptive. This is an important and unique feature that, for example, the LPC vocoders do not have. This would allow different wireless users the ability to choose different quality of speech, depending on their selection of cost. This would also allow the wireless suppliers to use the bandwidth effectively while satisfying their customer's requirements. This flexibility would be highly advantageous in the wireless market.

2.5.2 Results of Low Rank Error Modeling

To compare the actual and estimated error of the reconstructed signal, two equations will be developed. The predicted estimated error ratio is calculated by dividing the estimated MSE by the estimated reconstructed signal as

$$\frac{Eerror}{ESr} = \frac{\sum_{i=M+1}^{P} \sigma_i^2}{\sum_{i=1}^{M} \sigma_i^2} \tag{2.64}$$

Similar, the actual error ratio is calculated by dividing the actual MSE by the actual reconstructed signal as

$$\frac{Aerror}{ASr} = \frac{E\left[\| \mathbf{Error} \|^2\right]}{E\left[\| \mathbf{Sr} \|^2\right]} \tag{2.65}$$

In Eq. 2.65, the variables **Error** and **Sr** are both in hankel matrix form. These two equations both produce a single ratio for a given speech frame, for a specific number of parameters (M).

In Eq. 2.64, the unused singular values squared are divided by the used singular values squared. Again, the unused singular values represent the error and the used singular values represent the reconstructed signal. This method shows that the low rank error estimation method can be used to determine the number of parameters needed to represent a reconstructed speech signal. The more singular values are used, the smaller the error. A comparison of the ratio in both Eq. 2.64 and 2.65 are shown in Figures 2.15-2.18. Each Figure represents an individual frame of speech. The x axis represents the number of parameters (M) to reconstruct the signal, while the y axis represents the error ratio. Observe that the actual and estimate error ratio correspond nicely. Another observation, as expected, is that the error ratio goes to zero as the number of parameters used increases. The initial part of the x axis (the smaller values of singular values used) was omitted, since the actual error ratio is large due to the reconstructed signal being small. The reconstructed speech is small due to insufficient parameters to represent it. This combination increases the ratio, which makes the evaluation unstable in this region.

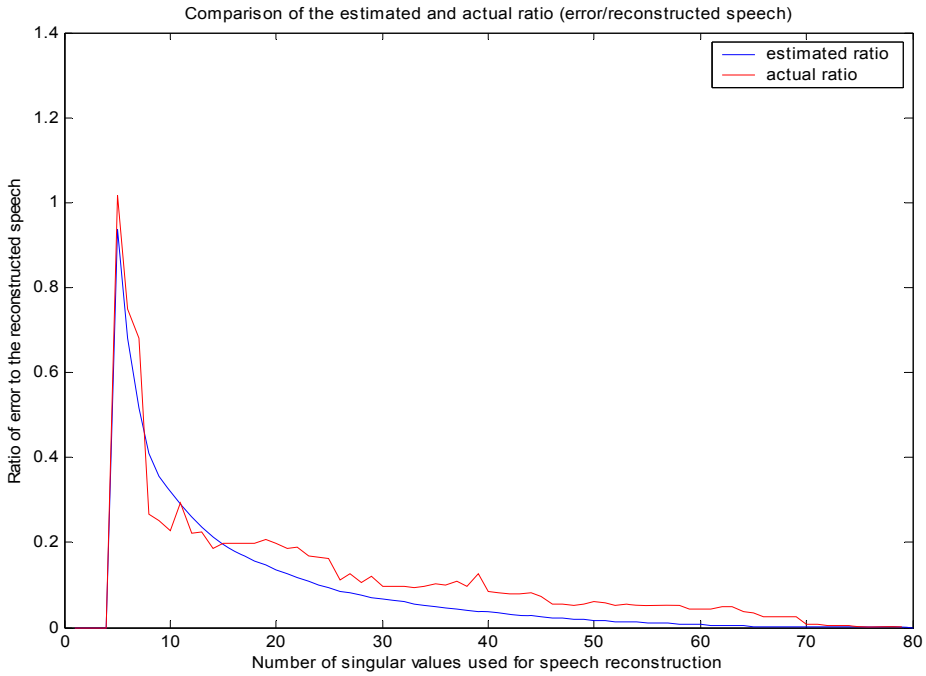

Figure 2.15: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. Note: Eerror/Esr (blue), Aerror/Asr (red)

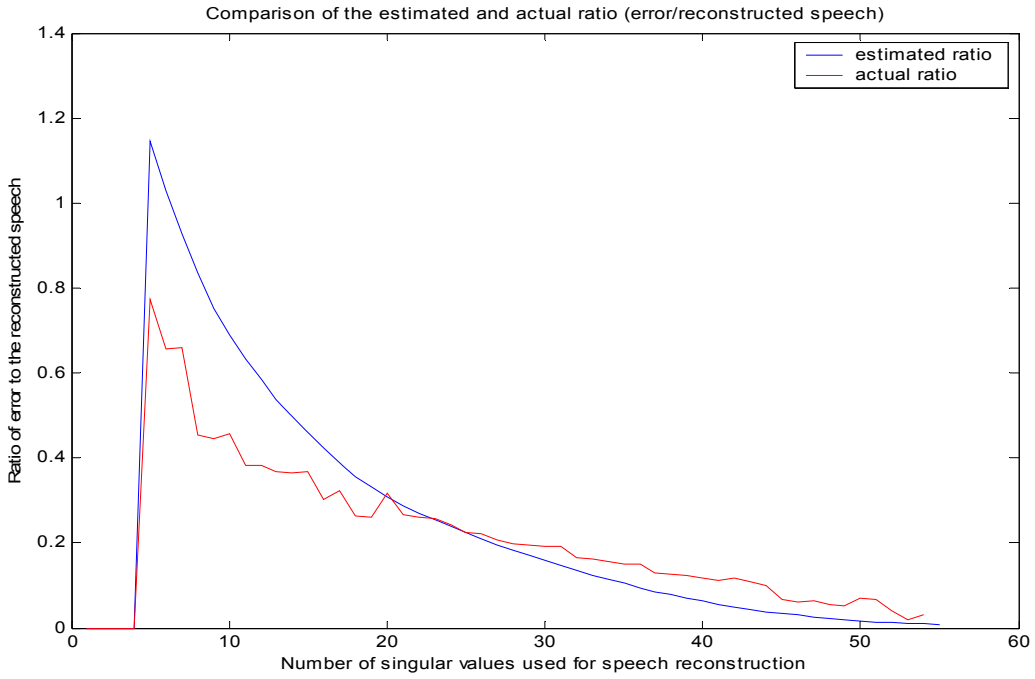

Figure 2.16: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. Note: Eerror/Esr (blue), Aerror/Asr (red)

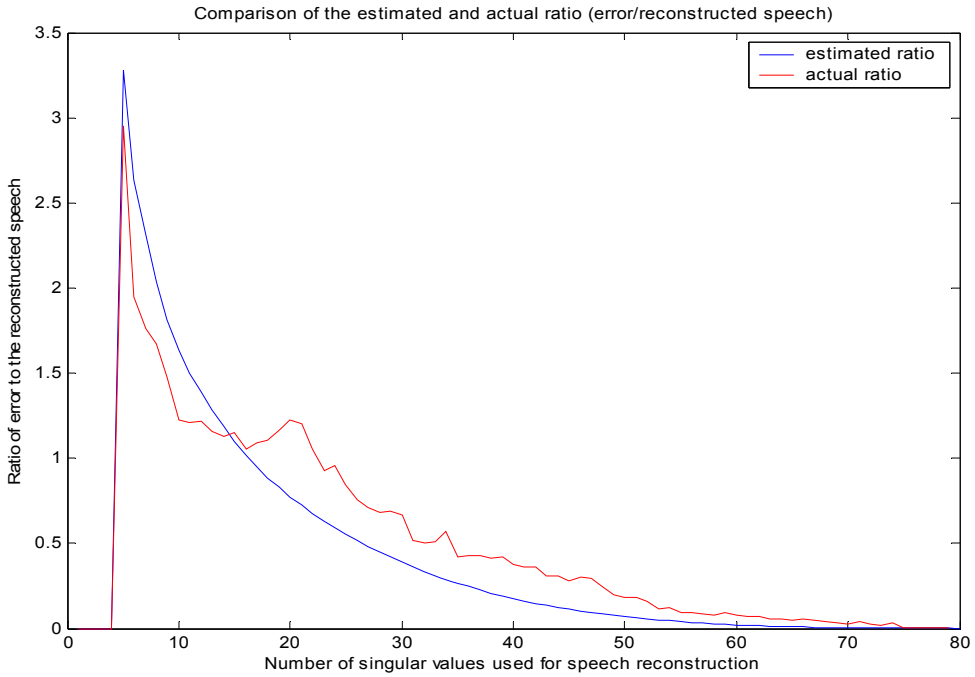

Figure 2.17: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. Note: Eerror/Esr (blue), Aerror/Asr (red)

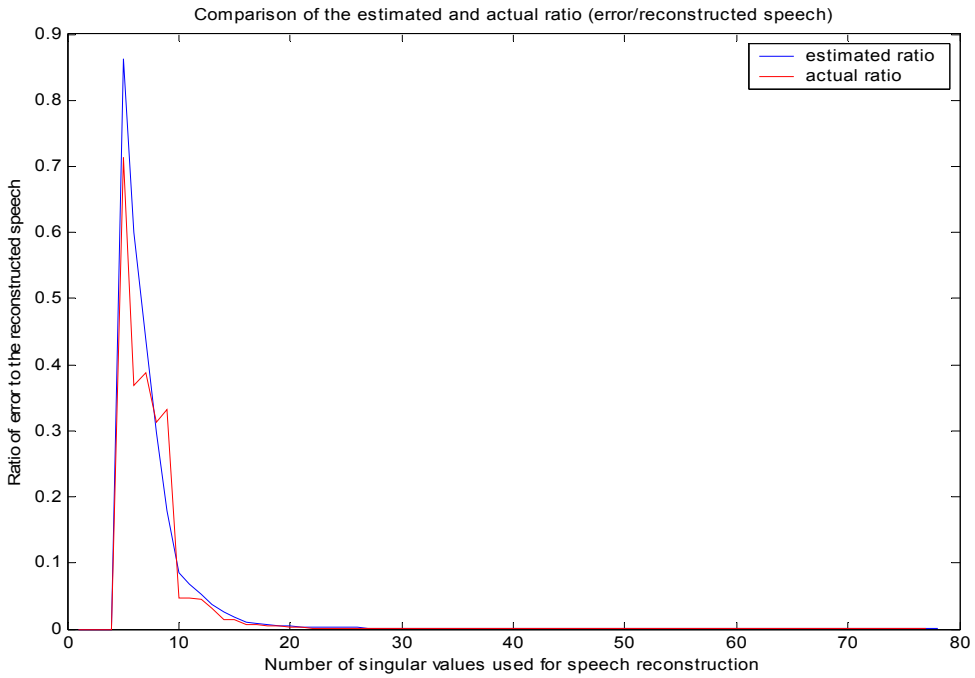

Figure 2.18: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. Note: Eerror/Esr (blue), Aerror/Asr (red)

Using the original signal as a baseline, rather than the reconstructed signal, is another good measure of error ratio worth discussing. This measure is similar to the above method with the exception of taking the ratio of the error to the original signal, rather than the reconstructed signal, as

$$\frac{Eerror}{ES} = \frac{\sum_{i=M+1}^{P} \sigma_i^2}{\sum_{i=1}^{P} \sigma_i^2} \tag{2.66}$$

and

$$\frac{Aerror}{AS} = \frac{E\left[\| \mathbf{Error} \|^2\right]}{E\left[\| \mathbf{S} \|^2\right]}. \tag{2.67}$$

The difference between Eq. 2.64 and 2.66, as well as in 2.65 and 2.67 is the denominator. In this case the denominator is the actual signal. In Eq. 2.66 the denominator is the sum of the all the squared singular values; all of the parameters represent the actual signal. The following plots, in Figure 2.19 - 2.22, show the curves of the calculated and actual error ratio. Similar to the first method, the actual and estimated error ratios correspond very well. This method shows the contour of the two error plots, which may be a better representation of the error ratio, especially in the low singular value regions. This is due to the fact that the signal for the lower portion of the x axis is much greater than the reconstructed signal, and therefore keeps the calculation stable.

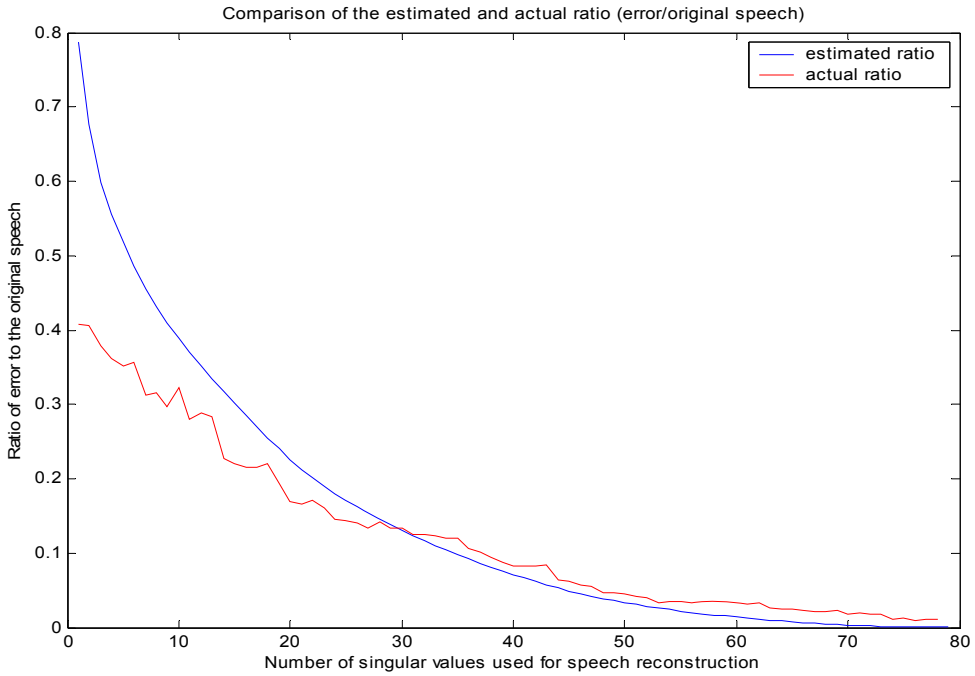

Figure 2.19: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. Note: Eerror/Esr (blue), Aerror/Asr (red)

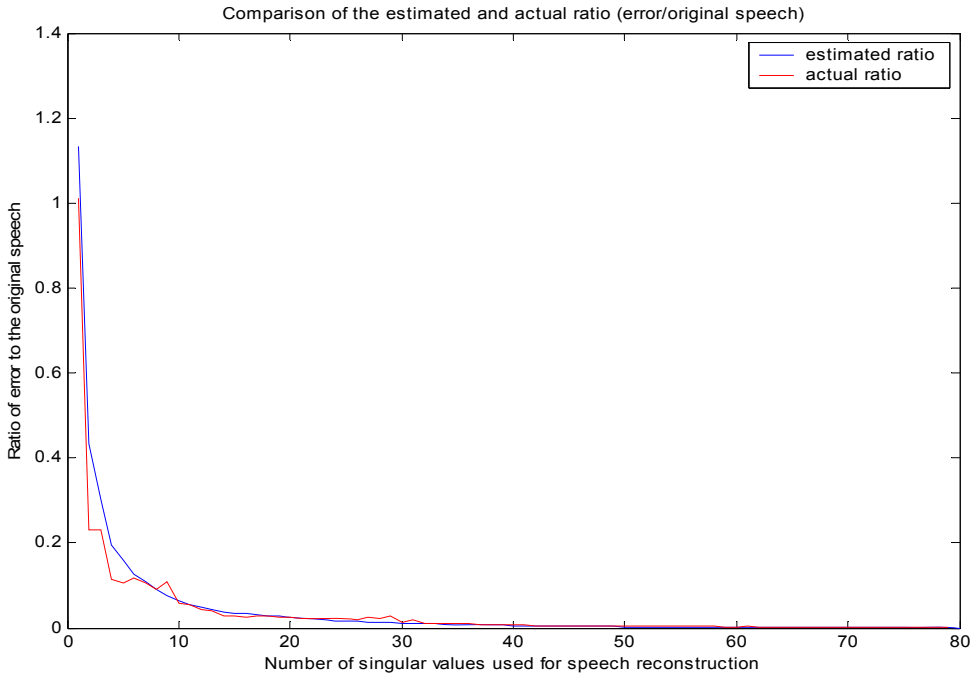

Figure 2.20: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. Note: Eerror/Esr (blue), Aerror/Asr (red)

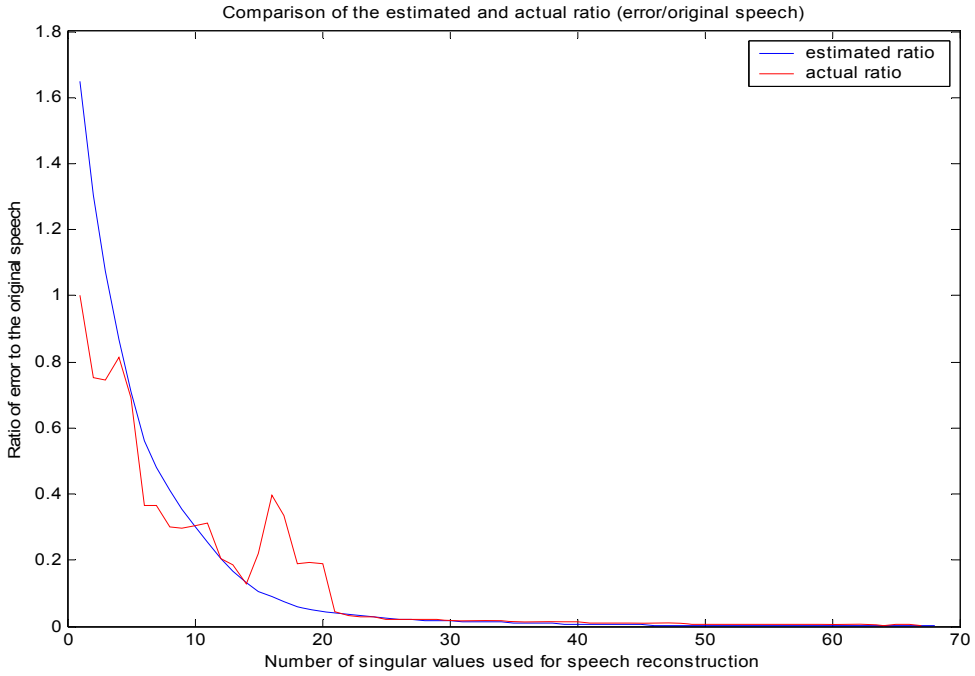

Figure 2.21: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. Note: Eerror/Esr (blue), Aerror/Asr (red)

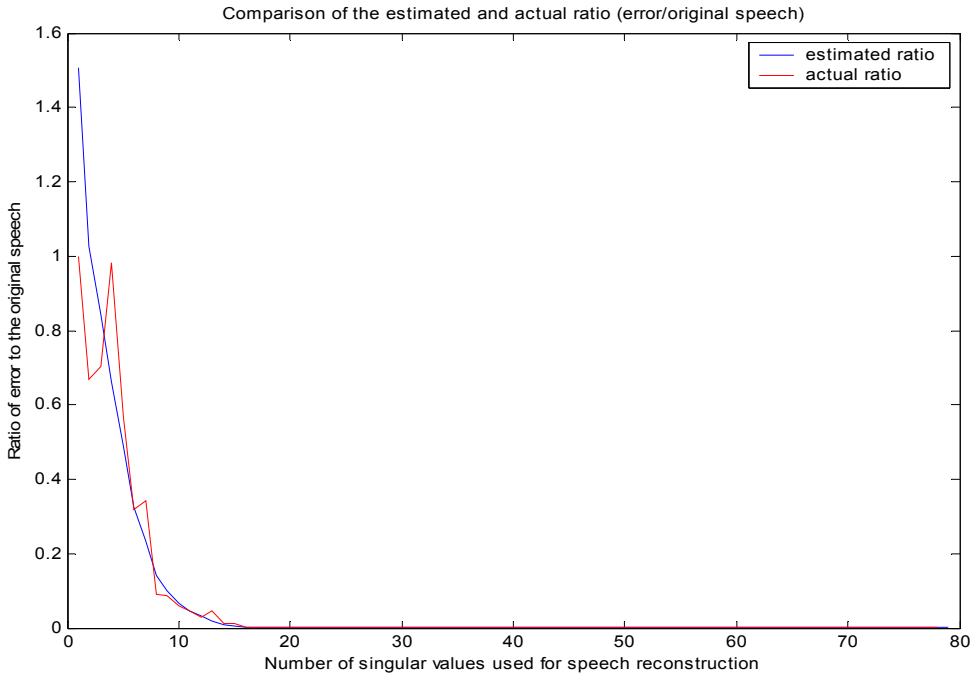

Figure 2.22: Comparing the actual error ratio to the estimated error ratio for a specific frame of speech. Note: Eerror/Esr (blue), Aerror/Asr (red)

To display the accuracy of the predicted/ estimated MSE relative to the actual MSE, a comparison of the ratio between the original and reconstructed signals were plotted. Figures 2.23-2.27 shows that the MSE can be predicted very well using the low rank error modeling estimation. The results show that the low rank error estimate is a very useful tool in determining the MSE prior to the pseudo-inverse in the MP approach. This is a strong benefit of the MP algorithm, particularly when processing time is critical in a lossy application, such as speech compression. In the next section, the MP is applied to the problem of speech compression.

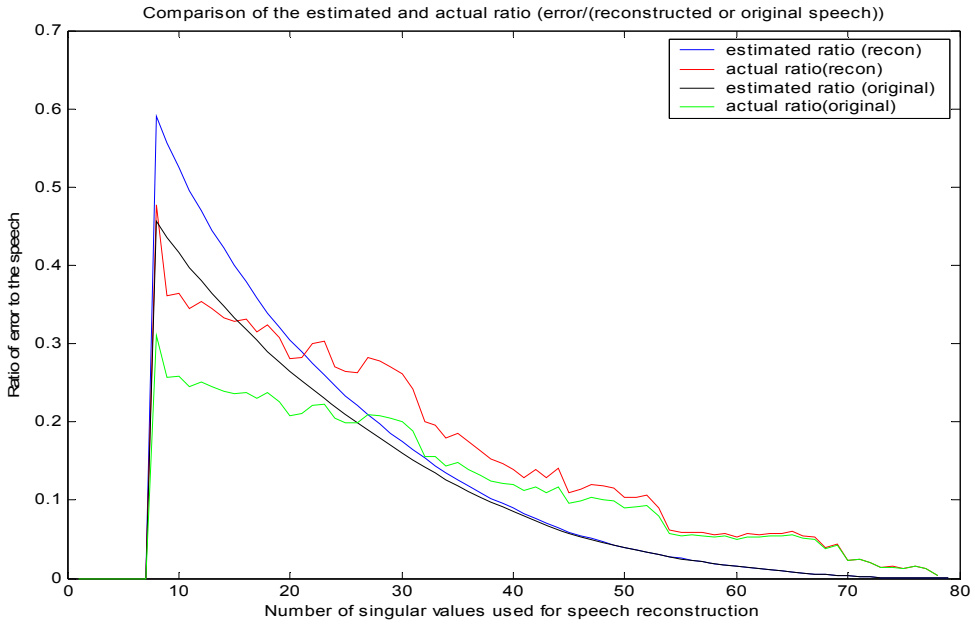

Figure 2.23: Comparing the error ratios using the reconstructed signal and the original signal as a baseline, for a specific frame of speech.
Note: Eerror/Esr (blue), Aerror/Asr (red), Eerror/Es (black), Aerror/As (green)

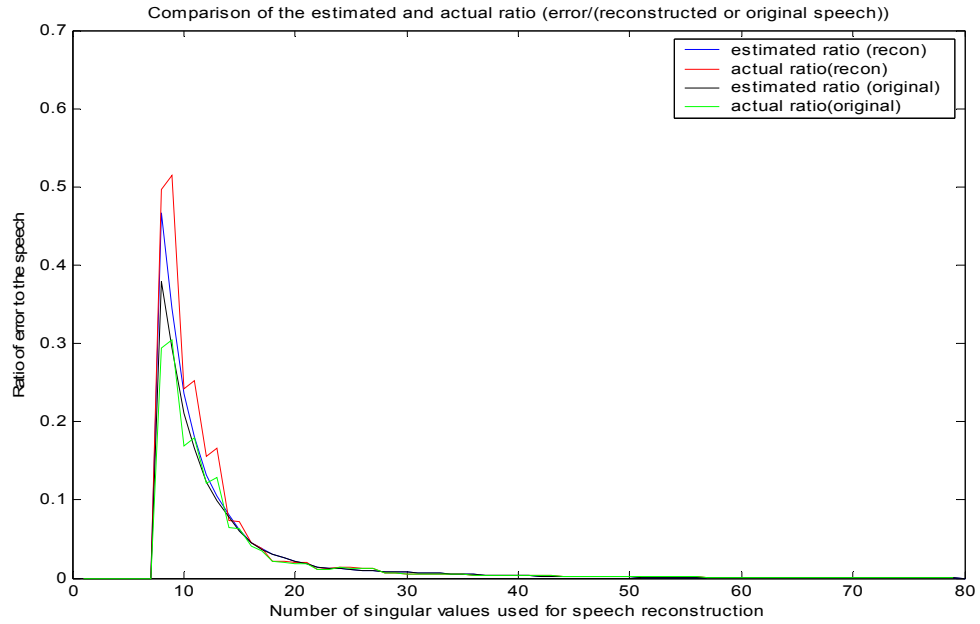

Figure 2.24: Comparing the error ratios using the reconstructed signal and the original signal as a baseline, for a specific frame of speech.
Note: Eerror/Esr (blue), Aerror/Asr (red), Eerror/Es (black), Aerror/As (green)

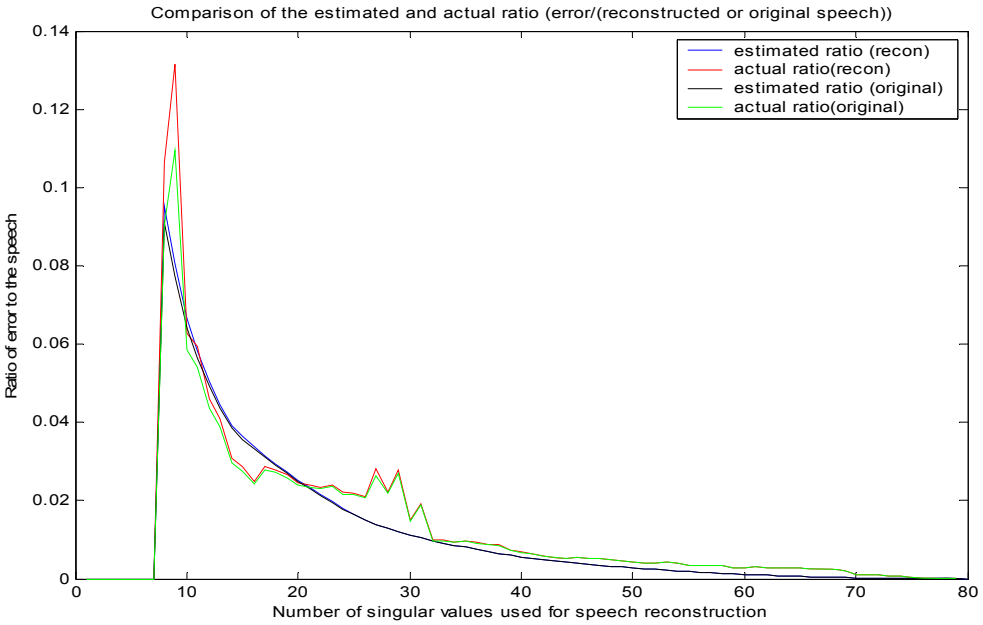

Figure 2.25: Comparing the error ratios using the reconstructed signal and the original signal as a baseline, for a specific frame of speech.
Note: Eerror/Esr (blue), Aerror/Asr (red), Eerror/Es (black), Aerror/As (green)

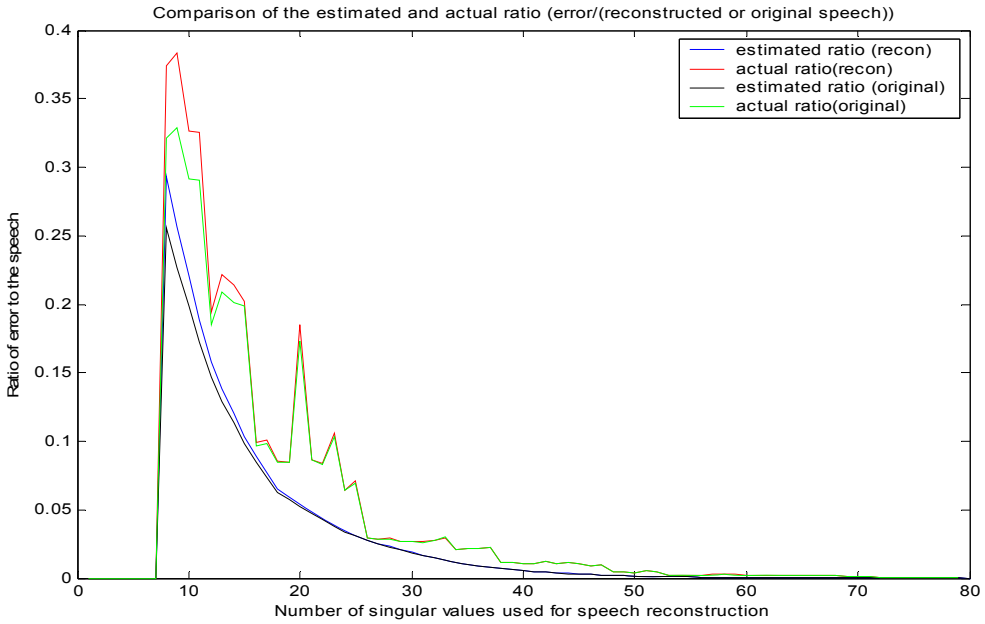

Figure 2.26: Comparing the error ratios using the reconstructed signal and the original signal as a baseline, for a specific frame of speech.
Note: Eerror/Esr (blue), Aerror/Asr (red), Eerror/Es (black), Aerror/As (green)

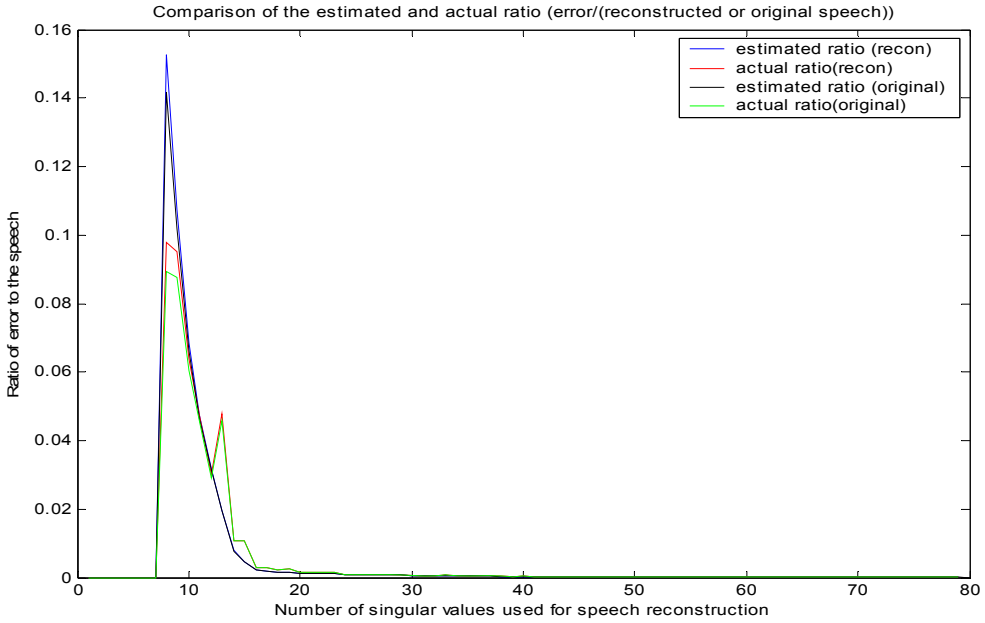

Figure 2.27: Comparing the error ratios using the reconstructed signal and the original signal as a baseline, for a specific frame of speech.
Note: Eerror/Esr (blue), Aerror/Asr (red), Eerror/Es (black), Aerror/As (green)

Chapter 3

Speech Compression Using the Matrix Pencil

3.1 Introduction

Speech processing is an important research area that has drawn much attention over the past few decades. Applications which utilize speech processing tools include, but are not limited to speech compression, speech enhancement, and co-channel interference reduction. These tools make use of a large variety of models and algorithms. The MP algorithm, which has been used mainly in phased array radar applications, is applied to speech compression in this chapter.

The compression of speech and audio signals can be found in many applications, such as cell phone, voice over IP, and digital voice storage to name just a few. Compression allows more users to use a system that would otherwise be available to just a few customers under constrained power or bandwidth resources. There have been many compression techniques developed in the past. Some techniques have a large compression rate with poor listening quality, while others have a small compression rate with good listening quality, and the others are between the two extremes.

In this research the focus is on compression of speech sampled at 8000 Hz. Typically 16 bits is used for uncompressed speech, giving a transfer rate of 128Kbits/sec. Speech compression results in data transfer rates in the range of 2.4kbps to 64kbps. At

2.4kbps the Linear Prediction Coder 10 (LPC10) vocoder has poor listening quality. Vocoders are classified as parametric coders, which have compressing rates of 53:1 to 8:1 with poor to good listening quality. Another type of signal compressor which can be used for speech, waveform coders, have a range of compression rates of 4:1 to 2:1 with good listening quality [20].

The MP speech compressor can be categorized as a waveform coder [37]. For MP based speech compression, the parameters of the MP algorithm can be quantized to 8 or 16 bits to help achieve significant compression rates. Experimentally we have observed compression rates of 10:1 with fair listening quality. Our present focus will be on comparison of other commonly used waveform coders up to 7:1 compression rates.

3.2 MP Speech Compression Algorithm

As previously discussed, a speech signal is first divided into segments often referred to as frames. In this experiment the frame lengths are 40 msec. long or 320 samples. This is longer than in the previous sections for the purpose of achieving higher compression ratios, while avoiding excessive speech dynamics, and still maintaining stationarity in any given frame. The frame of speech is used as in Eqs. 2.39 and 2.40. Using the unitary matrices, a specific number of poles are calculated via Eq. 2.46. This number depends on the targeted SNR and compression ratio. The speech frame can be voiced or unvoiced, and SVD aids in classifying which type it is**.** The singular values contained in the diagonal matrix $\mathbf{\Sigma}$ are dependent on the speech energy of the frame, and are in ranked order from largest to smallest. After normalizing each test signal recording prior to processing, a threshold of 1 was used in deciding the voicing class. Therefore,

the classification method was as follows: if the maximum singular value is less than 1 it is an unvoiced frame, otherwise it is a voiced frame.

The rank ordering allows one to reconstruct the speech signal using only a few dominant sinusoids. This feature is beneficial when the compression of speech is the goal, since the strongest poles will represent the reconstructed speech. By using a few dominant poles, and a weighted recombination of overlapping frames, the signal can be intelligibly reconstructed. The largest number of poles that can be used per frame is L; for this experiment L=160 corresponding to a frame size of 320 samples. With the exception of DC, these poles are complex conjugates, where a pair makes up a sinusoid. Each exponentially damped/ undamped sinusoid is made up of four parameters, amplitude, initial phase, decay, and frequency, for a maximum of 320 parameters per processed frame.

In this experiment, voice frames are assigned 15 sinusoids, i.e. M=30 poles. Unvoiced frames are assigned 4 sinusoids (M=8 poles). Each frame is overlapped 50% using a triangular weighted window during reconstruction; this smoothes the boundary of the frames, thus eliminating any discontinuity in the reconstructed signal. Each parameter of the sinusoid is quantized using either 8 or 16 bits. For voice frames a total of 60 parameters are used to represent the frame, while for unvoiced frames 16 parameters are used.

3.3 MP Speech Compression Results

For these results 190 recording files consisting of a mix of male and female speakers for a total of 38 speakers with 5 sentences for each speaker, were used from the

Texas Instrument / Massachusetts Institute of Technology (TIMIT) database [38]. The metrics used are the SNR, compression ratio and the processing time. SNR is defined herein as

$$10Log(\frac{P_s}{P_e}), \tag{3.1}$$

where P_s is the power of the original signal. The error power, P_e, is equal to the mean of the squared difference between the original and the compressed signal.

The first test examined the quantization of the input, either 16 bit PCM or 8 bit PCM. It also looked at the quantization of the parameters; frequency, phase, decay, and amplitude. They were quantized using 16 bit PCM, 8 bit PCM, or 8 bit mu-law. These quantized parameters were then used to reconstruct the output speech. Again, the number of sinusoids used to reconstruct the signal was 15/4 (15 for voiced, 4 for unvoiced). This was determined to be a good choice since the output signal was good sounding while giving a good compression rate.

The results, in Table 3.1, show that the 16 or 8 bit input speech does not impact the quality of the output signal, relative to the input. For the parameter quantization, it was found that the number of bits assigned to the frequency parameter needs to be 16 bits, while the remaining parameters should be quantized using 8 bit mu-law. This combination had the biggest impact on the results. Using method 9 as an SNR baseline, method 5 performed very well. The SNR was comparable to methods 2 and 3, but with an increase in the compression ratio. Method 5 requires that the frequency be quantized at 16 bits, while the other parameters are quantized at 8bit mu-law. Initial results were published in [39].

Table 3.1
SNR and Compression Ratio of MP Speech Coders

Speech Coder Method	**SNR (dB)**	**Compression Ratio**
1. Input=16 bit, MP 15/4, f=p=d=a=16 bits *	13.46	3.5:1
2. Input=16 bit, MP 15/4, f=d=16 bits, p=a=8 bits *	11.60	4.7:1
3. Input=8 bit, MP 15/4, f=d=16 bits, p=a=8 bits *	11.60	4.7:1
4. Input=16 bit, MP 15/4, f=16 bits, p= d=a=8 bits *	8.36	5.6:1
5. Input=16 bit, MP15/4, f=16 bits, p=d=a=8 bits mu-law *	13.01	5.6:1
6. Input=16 bit, MP 15/4, f=p=d=a=8 bits *	1.34	7.1:1
7. Input=8 bit, MP 15/4P, f=p=d=a=8 bits *	1.34	7.1:1
8. Input=8 bit, MP 15/4P, f=p=d=a=8 bits mu-law*	1.45	7.0:1
9. Input=16 bits, MP, no quantization	13.45	N/A

* f=frequency, d= decay, p=phase, a=amplitude; The number of bits each parameter is quantized

For the second test, the input speech files are 16 bit PCM sampled at 8 kHz. The signals were then compressed and quantized as described previously. The number of sinusoids used to reconstruct the signal was either 20 for a 2:1 compression rate, 15/4 for a better compression rate and a good SNR, or 18/1 for a better compression rate and a better SNR as indicated in Table 3.2.

The MP algorithm does not compare well to either the 8 bit linear PCM or 8 bit mu-law methods based on the SNR metric alone. However, it is shown that the MP method 5, performs very well when compressing the speech at a compression rate of approximately 5.2:1. This is a typical rate of a hybrid coder. According to the SNR the MP performs better than the equivalent ADPCM speech coder at the 5.3:1 compression rate (method 8). It also performs better than a DFT based algorithm, in which a subset of coefficients is retained. The DFT method was used as a comparison since it is an alternative method to the MP. In our implementation of the DFT-based compression method, the DFT size and the frame size are the same. The number of sinusoids per frame is fixed. Both variables are listed respectively in Table 3.2. Method 12 is

comparable to method 4 and 5 since the frame size and the number of sinusoids is similar when overlap is taken into account. Also, the MP amplitude and phase values are quantized similarly to the DFT amplitude and phase coefficients. Both use the mu-law method. The main difference is that the frequency locations for the DFT method need to be kept tract of during the transmission. An 8 bit word is used to transmit the DFT frequency locations of highest energies, limiting the DFT size to 512. The advantage of this method is that it is straight forward, easy to implement and easily runs in real time. (Real time refers to the sampling rate.)

Table 3.2
SNR and Compression Ratio of MP and other Speech Coders

Speech Coder Method	**SNR (dB)**	**Compression Ratio**
1. 8 bit linear	32.62	2:1
2. 8 bit mu-law	35.51	2:1
3. Input=16 bits, MP 20 sinusoids, f=p=d=a=16 bits *	15.34	2:1
4. MP15/4, f=16 bits, p=d=a=8 bits mu-law *	13.01	5.6:1
5. MP 18/1, f=16 bits, p=d=a=8 bits mu-law *	17.17	5.2:1
6. Input=16 bits, MP 15/4 no quantization	13.45	N/A
7. ADPCM 16kbps	12.05	8:1
8. ADPCM 24kbps	14.88	5.33:1
9. ADPCM 32kbps	19.72	4:1
10. ADPCM 40kbps	21.51	3.2:1
11. DFT, 256,32	12.22	5.33:1
12. DFT 320,41	12.62	5.2:1

* f=frequency, d= decay, p=phase,, a=amplitude; The number of bits each parameter is quantized

The quality of the MP compression at the specified number of sinusoids is good. The quality can be increased by increasing the variable M in Eq. 2.12, but the compression ratio is sacrificed. Thus the MP compression method is flexible with regards

to compressor ratios that can be accomplished. This can be seen in method 5, where the MP 18/1 method was used. Obviously the SNR was increased since more sinusoids were added in the voiced regions, the highest energy contributors. In all cases involving the MP, the intelligibility of the speech is very good, although the MP algorithm appears to add musicality to the signal. Musicality or tonality is the rapid coming and going of sine waves over successive frames [40]. Another benefit of the MP algorithm is that it eliminates the quantization noise, particularly when audibly compared to the 8 bit linear.

These experiments were run on a Pentium 4 2GHz processor with 1GB of memory running Windows XP using Matlab7.2. Whereas the DFT was able to run in real time, the MP algorithm does not when using this equipment. Depending on the number of sinusoids used to reconstruct the signal, the MP algorithm took at least 5 times real time to process using method 4 and 5. Whereas it took 7 times real time to process using method 3. This shows that the MP technique, in its present form, is expected to be capable of running in real time, with the next generation processor or with current real time hardware.

3.4 Conclusion

In this experiment a new speech compression algorithm using the MP algorithm to calculate speech parameters was introduced. A subset of these parameters could be used to reconstruct an estimate of the signal. To improve the listening quality when using the MP algorithm, methods to reduce or eliminate the musicality of the compressed signal is an option for future work. Other additional work still needs to be completed, such as looking at other metrics such as mean opinion score (MOS). The MP speech

compression method also can be implemented in real time floating point hardware in future efforts.

For compression purposes, poles were eliminated to represent speech with fewer parameters. In many scenarios, it is desirable to eliminate poles resulting from interferences such as tones that are present in speech. This topic is address in the next section.

Chapter 4

Band Focus Matrix Pencil Algorithm

4.1 Introduction

There are many different types of interferers and noise that can be inadvertently present in a speech signal. They range from crowd noise in a cocktail party, to wideband noise from a communication channel, to impulse noise from turning on a transmitter. In this dissertation tonal interference will be studied. Tones can easily contaminate speech signals by transducing from an electrical appliance such as a florescent light or a power supply. Tones can wreak havoc on communications systems, and speech processing recognition systems. Specifically, they can interfere with the process of correctly identifying a speaker, when the speaker identification (SID) algorithm is being applied. Without changes to the SID algorithm, the solution to this problem is to remove the tone from the speech signal.

Currently, the best removal solution is to apply the contaminated speech signal to a notch filter that is designed to remove the tone. The problem with a notch design is to determine the size of its bandwidth and location of the notch. Too small of a bandwidth will leave the tone in the speech signal. Too large of a bandwidth will remove the tone and some important speech information, which will leave a large "hole" in the spectrum. This can distort the speech signal prior to performing SID analysis. One way of measuring speech quality is by the root mean square error (RMSE). This determines how

well the tone removal process works. The RMSE is the error between the original and the enhanced signal. When a large notch filter takes too much speech information out, the RMSE will be high. It is very difficult to get an optimum filter due to dependency on the bandwidth and notch location. Depending on the data, the optimum bandwidth is always different. It will vary from speaker to speaker, or from frame to frame. Another problem with a notch filter is the group delay of the signal. The group delay is defined as the rate of change of the total phase shift with respect to angular frequency. The higher the group delay, the sharper the filter edge becomes, thus the output is improved. This luxury comes with a price. The sharper edges require a higher order polynomial for the filter. This is directly related to the number of taps. A filter will always have some group delay. The key is to keep that group delay to a minimum. This is difficult to do since there is tradeoff between edge sharpness and delay. It will be shown by applying the MP algorithm important speech information is retained that was otherwise removed by the notch filter. Interest in sub-band ESM signal representations is also evidenced in [1]

4.2 Stationary Tonal Detection

Prior to removing a tone it is necessary to first detect a tone within the speech signal. A tone can be identified by the SVD, and be removed by the MP algorithm. When a tone is present in a speech signal, it may have dominant poles depending on its strength when compare to the speech signal. Calculating the SVD of a frame of tone contaminated speech can identify the tones by the singular values.

4.2.1 Strong Tone Detection

If the tone is dominant the singular values that belong to the tone are also dominant with respect to the speech signal. If the signal has one tonal interferer, the number of dominant singular values is two, one for each component of a conjugate pair. When the poles of the tone are known or determinable, it is possible to reconstruct the speech signal without the tone interferer. Figure 4.1 shows the plot of singular values from the SVD of tone contaminated speech. In this frame, it can be seen that the first two singular values are dominant when compared to the enhanced speech. Notice the shift between the two plots, caused by the first singular values being represented by the tone, in the contaminated case. The dominance of the tone can be seen in Figure 4.2. The gain of the tone far exceeds the gain of the original speech.

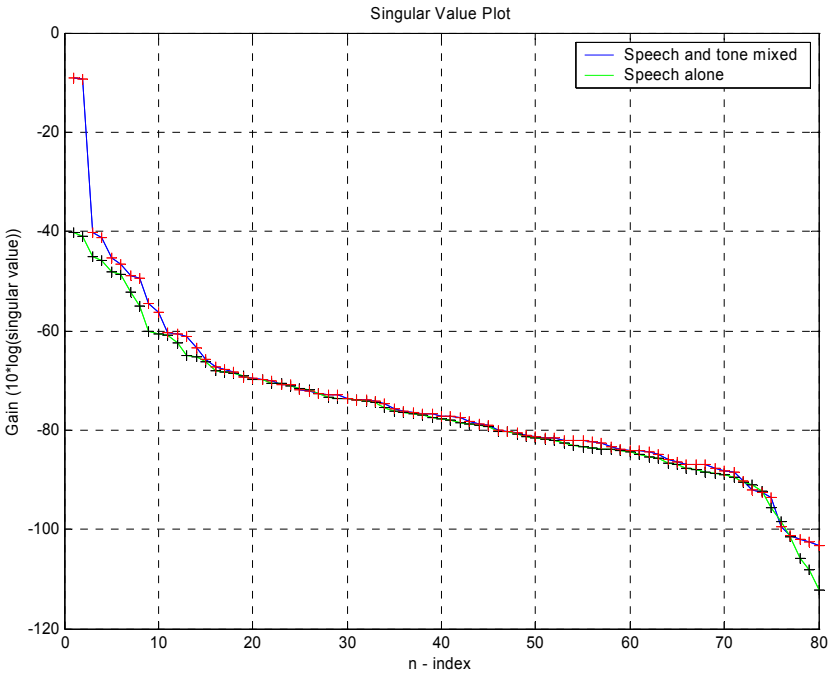

Figure 4.1: Singular Value plot, speech vs. speech + tone mixed.

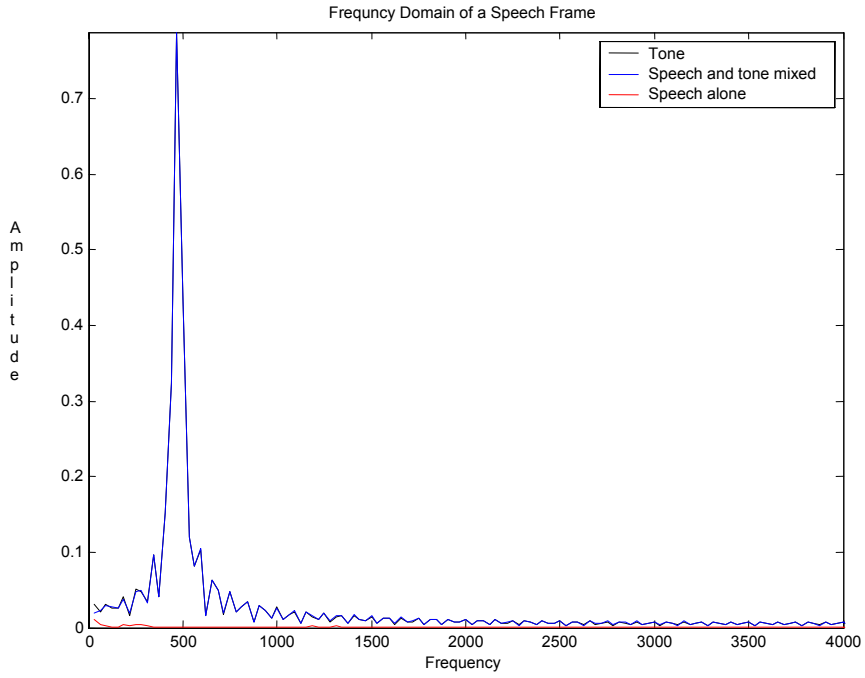

Figure 4.2: PSD of speech signal, tone signal, and speech + tone signal.

A detected strong tone can be regenerated alone without the speech, by performing the MP algorithm on the two dominant singular values; this is shown in Figure 4.3 and 4.4. In both plots the regenerated tone is nearly a duplicate of the original tone. Notice that the tone in Figure 4.4 has no decay or growth. In this example, the first two poles of the generated signal don't contain any information from the speech signal. This would occur when the interferer tone is much stronger than the speech, usually during silence or weak to moderate unvoiced frames.

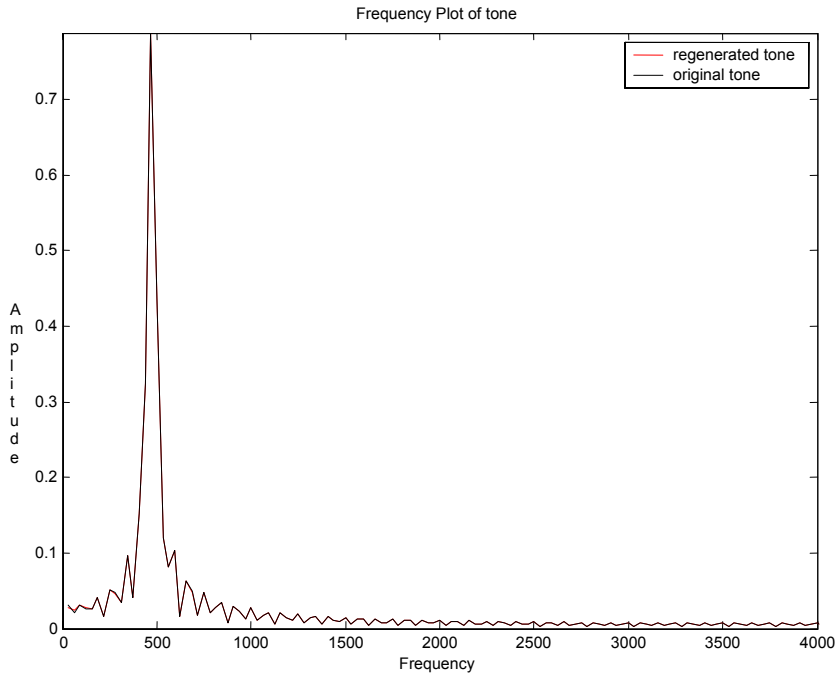

Figure 4.3: PSD comparison of original tone and reconstructed tone signal.

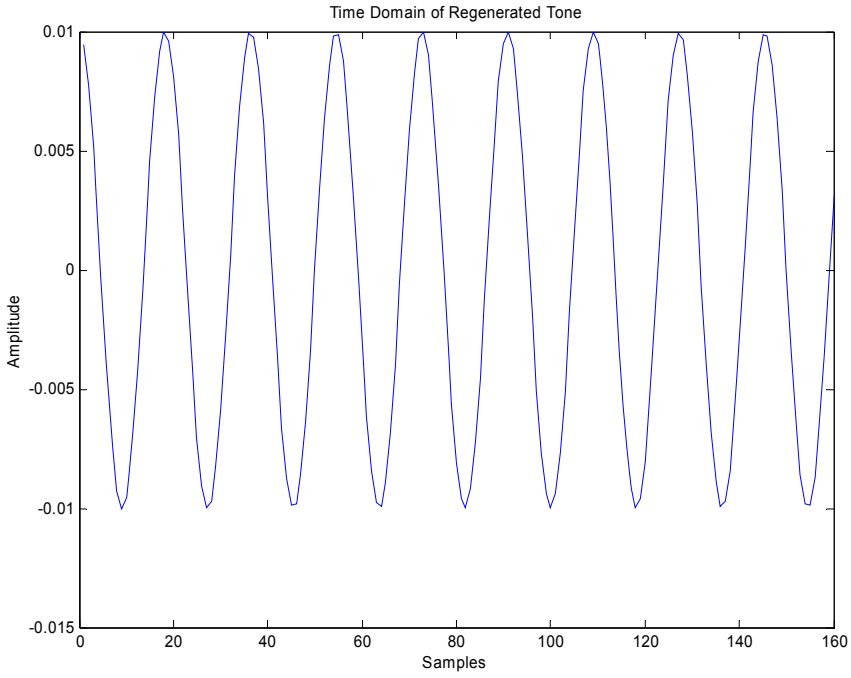

Figure 4.4: Time domain plot of the reconstructed tone signal.

4.2.2 Moderate Tone Detection

In the case in which the singular values for the tone are not as dominant, the reconstruction of the tone is not a perfect match of the original tone. An example is shown in Figure 4.5, where the strong singular values are displayed. The first two values of the speech and tone mixed are slightly different to the values of the speech alone. Also, notice that the 3rd and 4th singular values for the speech and tone mixed are the same as the singular values for the 1st and 2nd in the speech alone. This is true for all larger singular values. This shift indicates the first and second singular values in the speech and tone mixed signal are affected by the poles of the tone. In this example the tone's amplitude is on the order of the speech amplitude, thus the tone's dominance is not as clear for this frame.

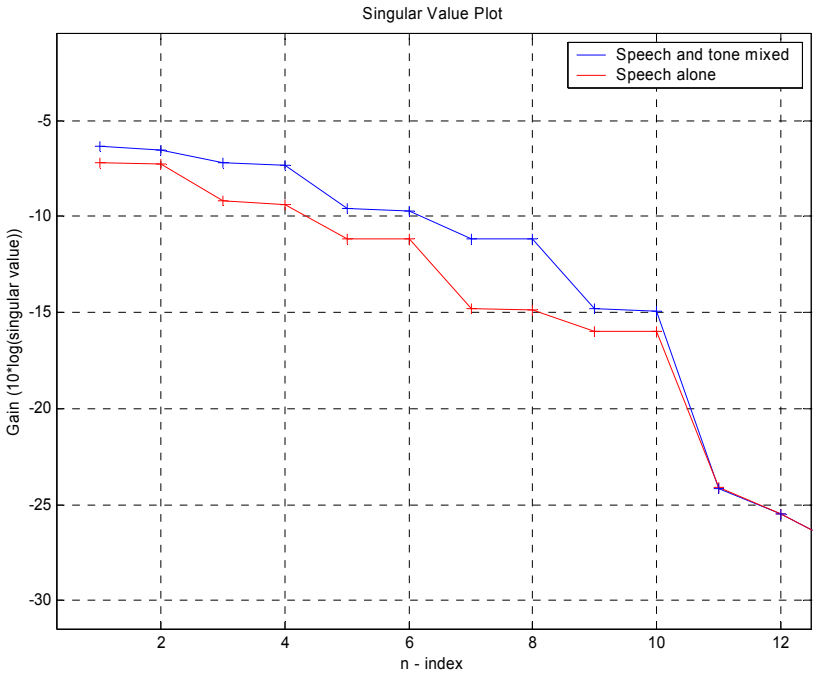

Figure 4.5: Singular Value plot, speech vs. speech + tone mixed.

The power spectrum density plot of the tone, the speech, and the speech and tone mixed are shown in Figure 4.6. There are two additional observations one should note. First, the speech frequencies that have high gain are lower than the tone; this can be an indication of a voice frame. Secondly, the speech and tone mixed signal at 400 Hz has higher amplitude than the tone signal alone. This is caused by the addition of speech and the tone for that particular frequency. When reconstructing the signal with 2 poles, shown in Figure 4.8, a decaying factor is introduced to the reconstructed tone. This damping portion is introduced from the speech. It can then be assumed that the noise space and the signal space intersect. Therefore, in this frame, it is very difficult to remove the tone without slightly distorting the speech.

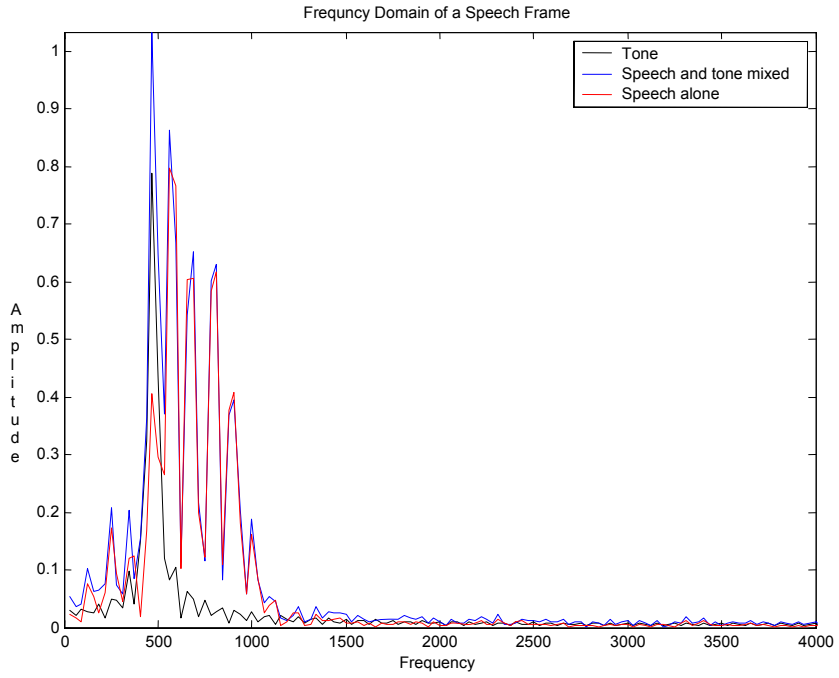

Figure 4.6: Power spectrum density comparison of a speech frame.

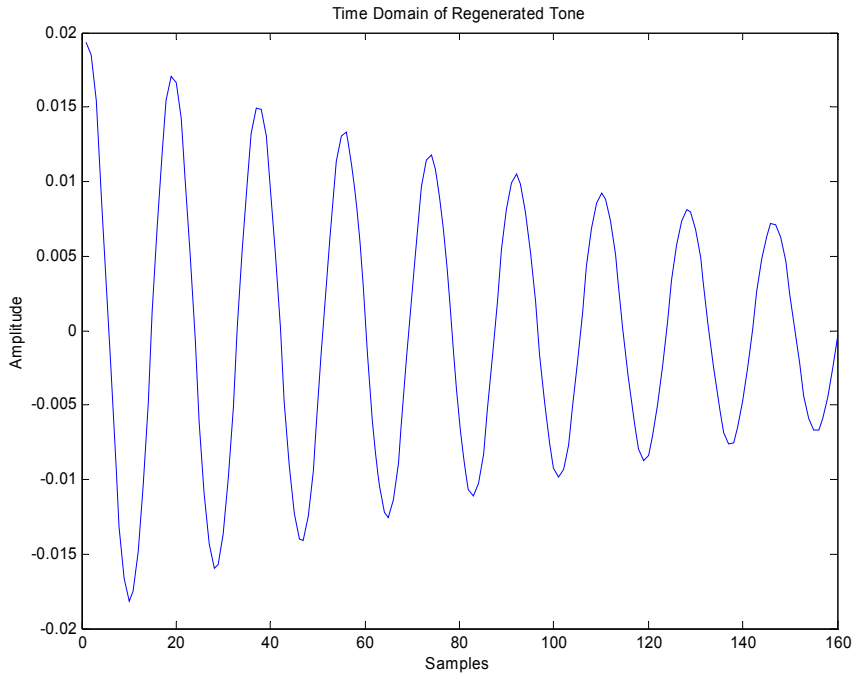

Figure 4.7: Time domain of reconstructed signal.

4.2.3 Weak Tone Detection

In the next case, the speech signal is stronger than the tone, as seen in Figure 4.8 and 4.9. Figure 4.8 shows that the first two singular values, of both signals, are the same. This indicates that these are speech related poles. The next pair for the speech + tone case represents the tone. Notice the 3rd and 4th values of the speech alone equal to the 5th and 6th values of the speech and tone mixed, and so on. This shift is further evidence that the poles of the tone are the 3rd and 4th singular values for the speech + tone case. To prove that the tone is placed at the 3rd and 4th pole, Figure 4.10 and 4.11 show the PSD plot of the generated tone by using these poles. In Figure 4.10, 4 poles were used to reconstruct the signal; while in Figure 4.11, 2 poles were used. It can be seen that the tone is regenerated when the 3rd and 4th pole are used in the regeneration of the signal. This

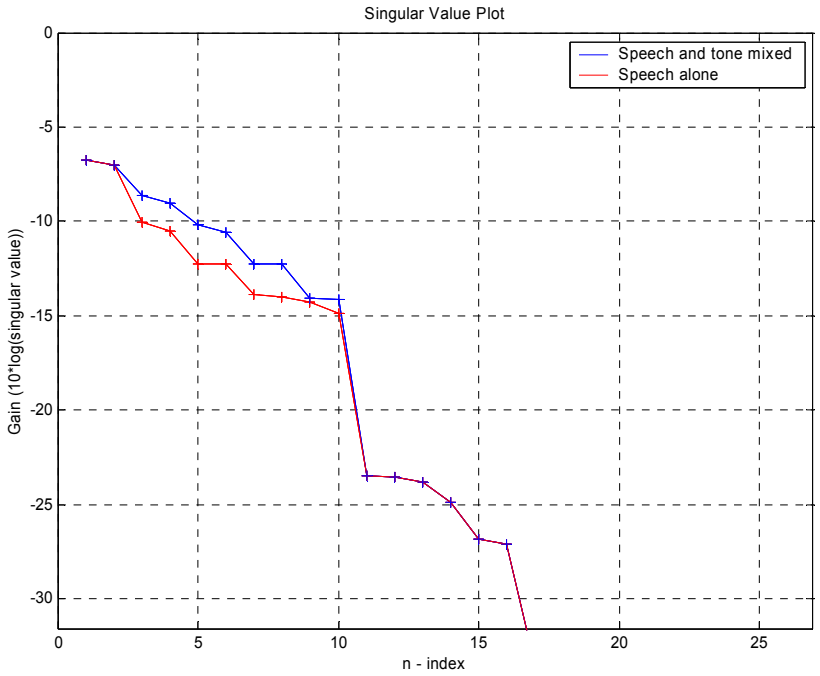

Figure 4.8: Singular Value plot, speech vs. speech + tone mixed.

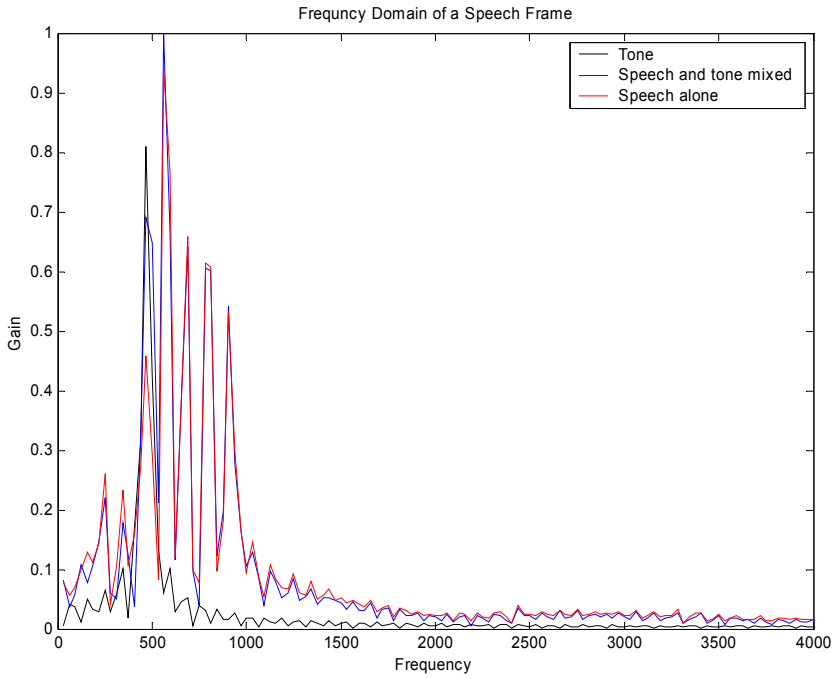

Figure 4.9: Power spectrum density comparison of a speech frame.

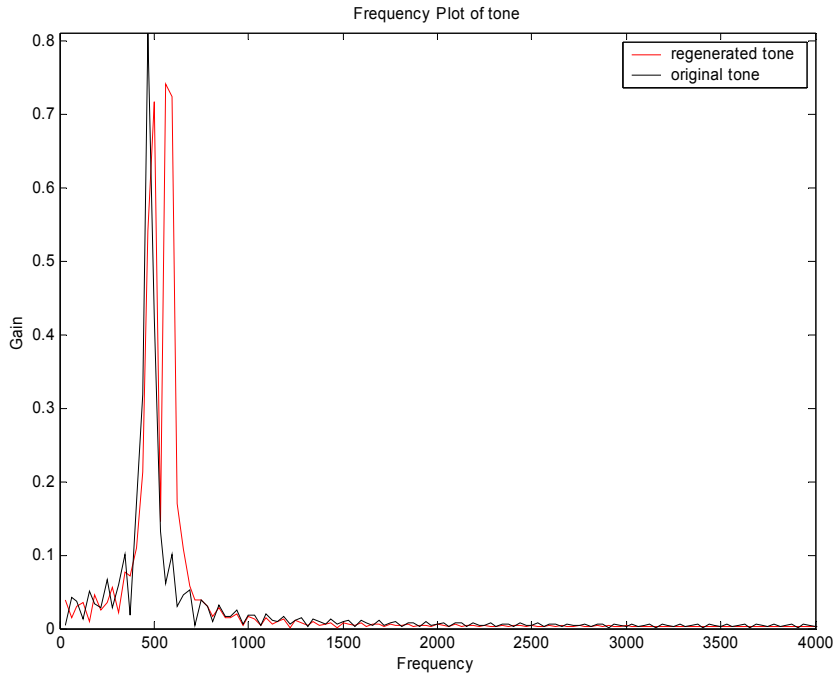

Figure 4.10: Power spectrum density comparison of a speech frame.

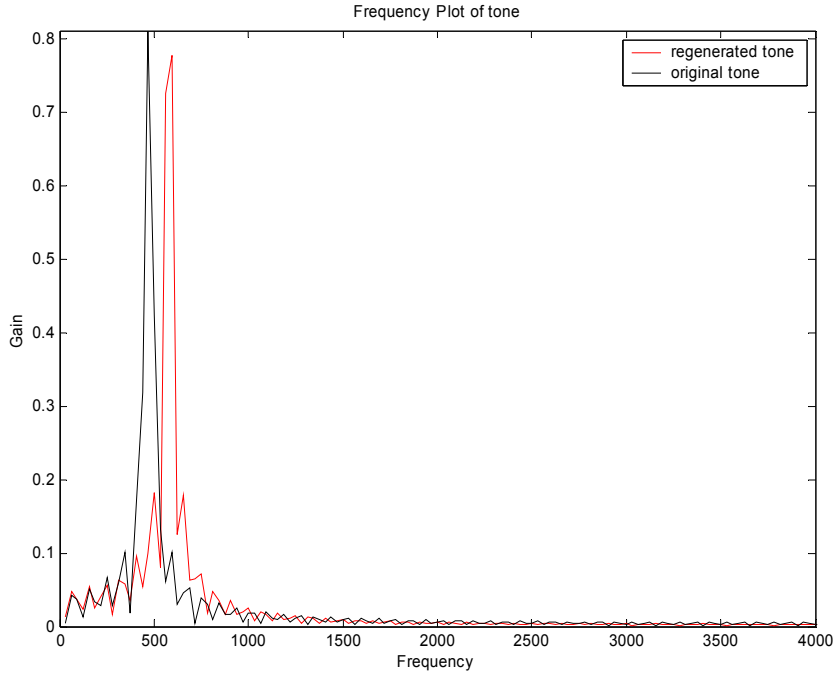

Figure 4.11: Power spectrum density comparison of a speech frame.

implies that the speech poles at 600 Hz are represented by the 1st and 2nd poles of the speech + tone signals. Notice the shift of the 400Hz signal in Figure 4.10. This is caused by the mixture of speech information with the tone, similar to the signal shown in Figure 4.6.

4.2.4 Stationary Tone Detection Conclusion

These experiments and observations demonstrate that the SVD can be used to detect a stationary tone in an unvoiced or silence speech frame, representing a strong tone scenario. It was observed that the singular values of a frame of speech represent its signal space. When the tone is completely dominant, the signal space vectors of the tonal interferer are completely orthogonal to the signal space of the speech signal. While other times, when the speech and tone are similar in amplitude, they are non-orthogonal. It was shown that this orthogonally depends on the strength of the speech and the interferer tone. Therefore, the SVD can be used to detect stationary interfering tones during frames that consist of silence or unvoiced signals.

The procedure for the detection of an interferer stationary tone via the SVD method is as follows. For each frame the slope between the second and third singular value is calculated. If the slope is above a threshold, then the frame is chosen as a target frame. These target frames are very likely to be an unvoiced frame or a silence frame, since the energy of the tone is much larger than the energy of the speech. They can also be low energy voiced frames, in which the tone energy is larger than the speech energy. Once these target frames are chosen, the next step is to determine the pole of the tone. This is done by completing the MP algorithm on the targeted frames, with the number of

poles equal to 2. Once the first two poles of these targeted frames are known, the next step is to determine the frequency of the tone. This is accomplished by taking a statistical approach. For each targeted frame, the frequency component of the pole is placed in 10 Hz bins. If the bandwidth of the signal is 4 kHz, there would be 400 bins. The bin with the largest number of placements is chosen. The actual frequency is the value in the middle of the bin. Once the value of the frequency is determined the damping factor is assumed to be 0. The targeted pole is now known, and the tone removal process follows. Currently this method is a non-real time process applied to recorded speech signals.

4.3 Tone Removal

Once the frequency of the tone is detected, it would need to be removed. The tone can be removed in multiple ways. In this research various ways were developed. Two methods particularly worked very well. These two methods are called the Band Focus Matrix Pencil Temporal Subtraction (BFMP-TS) and Band Focus Matrix Pencil Temporal Subtraction (BFMP-TR). The other methods that were used, but were not as successful, were called Zero Poling and Temporal Subtraction. In Zero Poling the pole of the tone is zeroed out, and in Temporal Subtraction the tone only is regenerated and then subtracted from the speech signal.

The Zero Poling method is not discussed further. The Band Focus methods were studied with regard to performance and compared to a notch filter. The metrics used in the comparisons were the RMSE and SID. For SID results, the recognition system is trained on original speech but tested using contaminated and cleaned speech.

4.3.1 Notch Filter Baseline

The notch filter technique is used quite extensively to remove tones. A notch filter does a fine job in taking out a tone [20,41]. Although, the notch filter introduces a group delay and the bandwidth needs to be optimized. One can design a notch filter with a small bandwidth such that only a small amount of speech data is attenuated or removed. However, during a practical communication setting, it is difficult to determine the optimum bandwidth of the notch. Too small of a bandwidth and the tone will still be in the signal. Too big of a bandwidth and it will take out too much speech information along with the tone. The optimum bandwidth depends on the data, so for a particular signal, the bandwidth should ideally vary from frame to frame. Another problem with a notch filter, or any type of filter, is the group delay. For a symmetric finite impulse response (FIR) filter, the group delay is dependent on the number of taps or the variables in the polynomial of the filter. If the filter has more taps, the steeper the filter frequency response edges. This causes a bigger group delay. Otherwise, with a small number of taps the filter frequency response edges have a gentle slope. This produces a shorter group delay. For FIR filters, the group delay is the number of taps minus one divided by two, and is the number of samples the signal is delayed after filtering. This produces a tradeoff between the slope of the filter edges and the size of the group delay, when signal delay is a concern,

It will be shown that the Band Focus Matrix Pencil (BFMP) overcomes the notch filter's delay tradeoff. A filter with a low number of taps can be used along with the MP algorithm, and provide results of a filter with a high number of taps. This would lower

the group delay, and still provide the desired results of a filter with steep edges. It will be shown in the result section, how the notch method compares to the MP techniques produced in this research.

4.3.2 Band Focus Matrix Pencil Temporal Subtraction

The BFMP-TS goal is to reconstruct the tone only by using the MP method and then subtracting it from the original signal. The contaminated speech signal, with unity gain is band pass filtered to allow the tone and some speech information to pass to the MP. This allows the MP algorithm to only process the spectrum that is around the tone. This allows the MP algorithm to ignore the remaining spectrum, thus "focusing" on the region of interest. Once the two conjugate poles of the tone are located, the tone is then reconstructed by zeroing out all of the other poles related to the band passed speech. To zero out the poles, the amplitude of the respective pole is simply set to zero. Once this is achieved, the tone can be subtracted from the contaminated signal. This would be done for each frame. The block diagram of the BFMP-TS is shown in Figure 4.12. The band pass filter is designed as an FIR filter with an integer group delay. The delay element in Figure 4.12 is set to this same delay value for proper signed alignment.

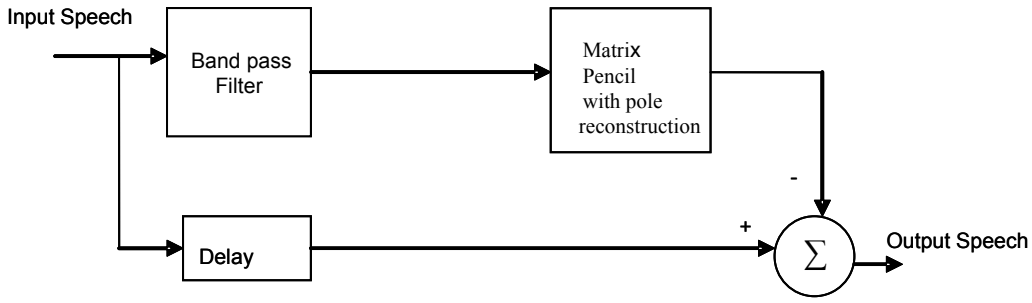

Figure 4.12: BFMP-TS algorithm: Matrix Pencil combination with band pass filter to subtract the interfering tone.

The original signal prior to the any contamination of the tone is shown in Figure 4.13. A 400 Hz interferer tone was mixed with a speech signal, shown in Figure 4.14. The resulting contaminated signal was inputted into the BFMP-TS algorithm. The output signal's spectrogram is shown in Figure 4.15. Notice the slight residue of the tone at 400 Hz. This residue is caused by the distorted tone estimate as described previously in section 4.2. Although the tone has been reduced, the tone residue has a slight impact on the RMSE and the SID results. These results are shown in section 4.3.4. Also, one can see the difference between this spectrogram and the spectrogram of the notch filter results, shown in Figure 4.16. The notch filter puts a hole in the spectrum. This notch filter approach takes out significant amounts of speech information along with the tone.

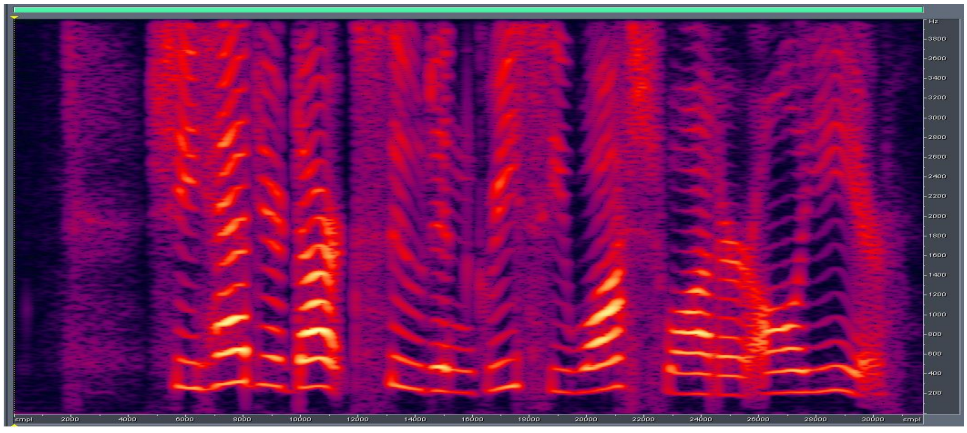

Figure 4.13: Spectrogram of the original audio signal
Note: x-axes is in samples (0-32000), y-axes is in frequency (0-4kHz).

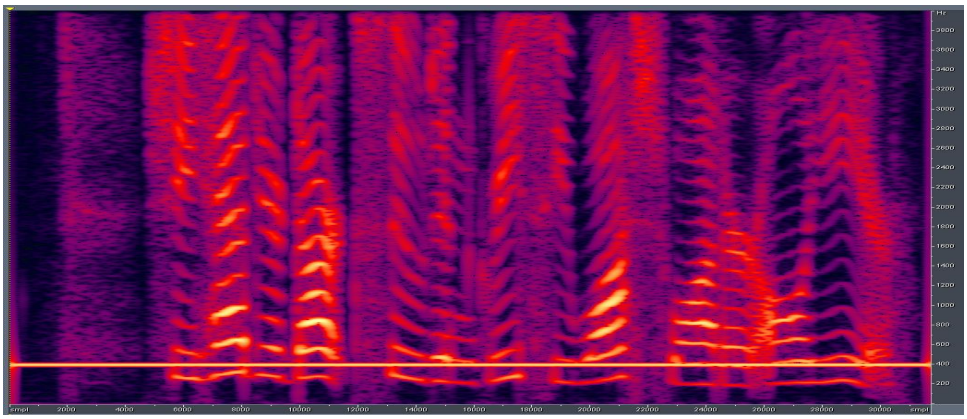

Figure 4.14: Spectrogram of the original audio signal mixed with a 400 Hz tone
Note: x-axes is in samples (0-32000), y-axes is in frequency (0-4kHz).

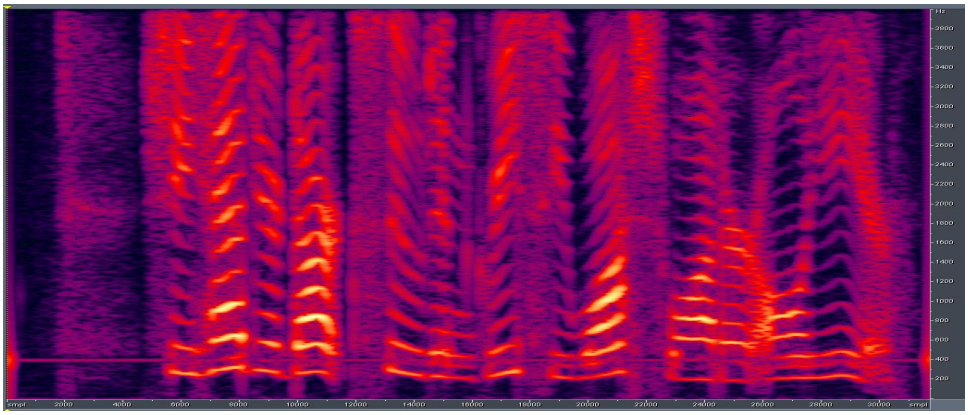

Figure 4.15: Spectrogram of an audio signal after BFMP-TS
Note: x-axes is in samples (0-32000), y-axes is in frequency (0-4kHz).

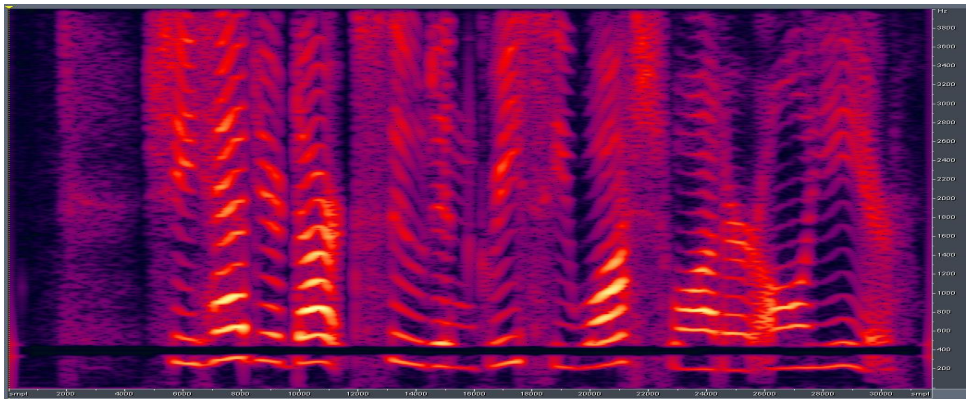

Figure 4.16: Spectrogram of an audio signal after a 15% notch (15% of the tone frequency, 400 Hz) Note: x-axes is in samples (0-32000), y-axes is in frequency (0-4kHz).

4.3.3 Band Focus Matrix Pencil Temporal Reconstruction

Another method that was developed is called the BFMP-TR. In this method shown in Figure 4.17, the band pass filter takes out the unwanted tone and the surrounding speech. This part of the spectrum is subtracted from the original contaminated and delayed speech signal. The result at the output of the first summer is a clean signal minus the band pass spectrum that was removed. This result is similar to a contaminated signal processed through a notch filter. The problem with this signal, as one can listen to it, is the "tunnel" effect on the quality during listening test. This can be compared to the physical scenario when someone speaks through a tunnel; there is an unnatural quality to it. The unnatural sounding speech is caused from the hole in the spectrum. To solve this problem, the MP algorithm is used to reinsert the speech information, which was removed from the subtraction of the band pass signal, back into the enhanced speech. The idea is to reinsert the speech, but not the tone. The method looks for any poles that are close to the detected tone. In the z-plane, the distance of each pole is calculated from the detected pole using an Euclidian measure. All poles within a specific threshold of the detected pole are zeroed out. This is done by zeroing the pole's respective amplitude. The MP algorithm then reconstructs this part of the signal with the zeroed out poles. The output of the MP algorithm then consists of only vital speech information, which is then added to the signal from the output of the first summer. This process in effect fills in the spectral hole. This can be seen by comparing the spectrograms in Figure 4.18 and 4.15. Also, when Figure 4.13 is compared to the Figure 4.18, one can observe the similarities of the BFMP-TR method to the original spectrogram. This method eliminates the tunnel effect, giving a more natural sounding

speech signal. It also leaves a smaller amount of tonal residue. The reason for this is because the signal reconstruction and any associated errors are spread across a wider bandwidth. It also has other benefits, including a small group delay and robustness when compared to a notched filtered signal. These results will be discussed in the result section.

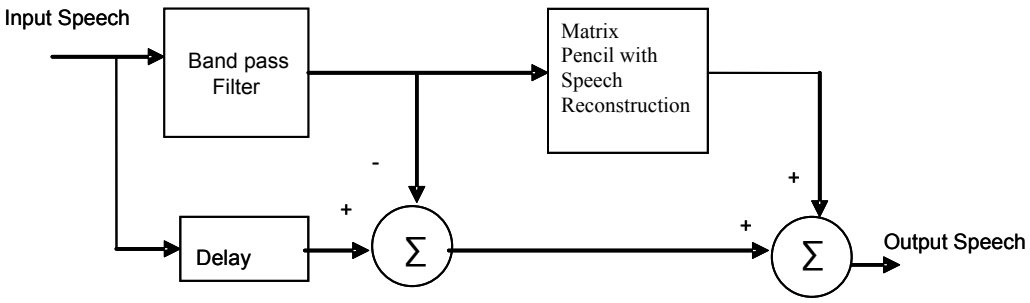

Figure 4.17: BFMP-TR algorithm: Matrix Pencil combination with band pass filter, to reconstruction speech while removing the interfering tone.

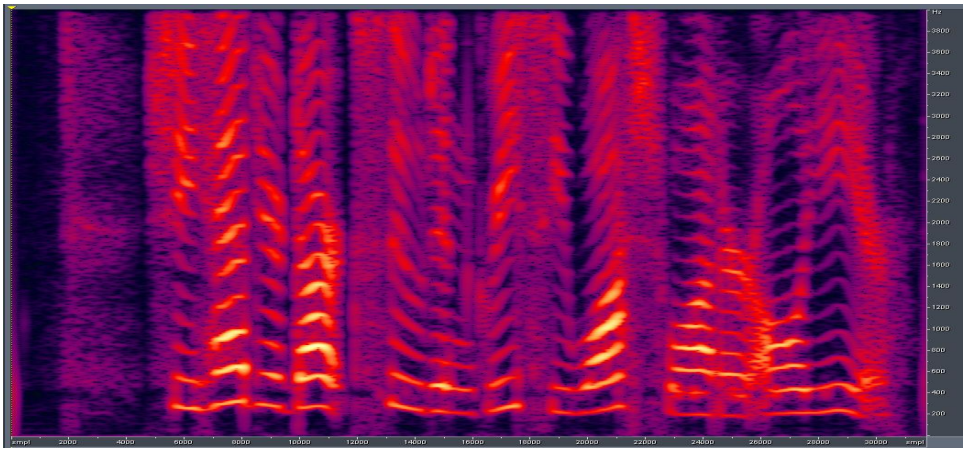

Figure 4.18: Spectrogram of an audio signal after BFMP-TR
Note: x-axes is in samples (0-32000), y-axes is in frequency (0-4kHz).

4.3.3.1 Cascade BFMP-TR

A common problem with electronic devices is that they can create harmonics of the 60 Hz power line in the audio signal. Harmonics are the by-products of modern electronics, which include personnel computers, uninterruptible power supplies, variable frequency drives (AC and DC) or any electronic device using solid state power switching supplies to convert incoming AC to DC. These harmonics can interfere with communication systems that they are connected to [42]. The IEEE 519 Standard addresses this problem and presents solutions. If the IEEE Standard is not being applied, the result could be tonal interference within a speech signal. This interference would be at the 60Hz frequency and its harmonics, either odd and/or even. Once this interference is embedded in the speech signal, it is very difficult to remove. This interference is more commonly known as a "hum". Power line "hum" can have a large impact on SID processes, because it has closely spaced harmonics that spread throughout the speech spectrum.

This BFMP-TR method can be applied to this type of contaminated speech. An enhancement process can have a difficult time in cleaning a speech signal with multiple tones. Therefore, a cascade methodology was applied to the BFMP-TR method to tackle this type of problem. A block diagram of this system is shown in Figure 4.19. In this cascade system, the user can insert as many stages as there are tone regions. Each stage would be dedicated to the region around a tone. A band pass filter would pass the tone and its surrounding spectrum to the MP algorithm. The MP algorithm would remove the tone and output the speech that was removed from the band pass filter. The band pass signal is also subtracted from the delayed signal as before. This signal is fed to the next

stage and the process is repeated. The signal coming out of the last stage (signal y_N) is equal to the input signal minus all the band pass signals. The final output signal is equal to signal y_N plus signals y_1, y_2 … y_M. This output signal is the contaminated signal without any of the tones. In practice, only a few cascade stages would normally be needed. The delays are necessary (Delays 1,2,3,..M) to synchronize the output signal. Note that delay M is equal to no delay, since y_m and y_n are synchronized. To avoid some of the delays, there may be more efficient ways to design this concept.

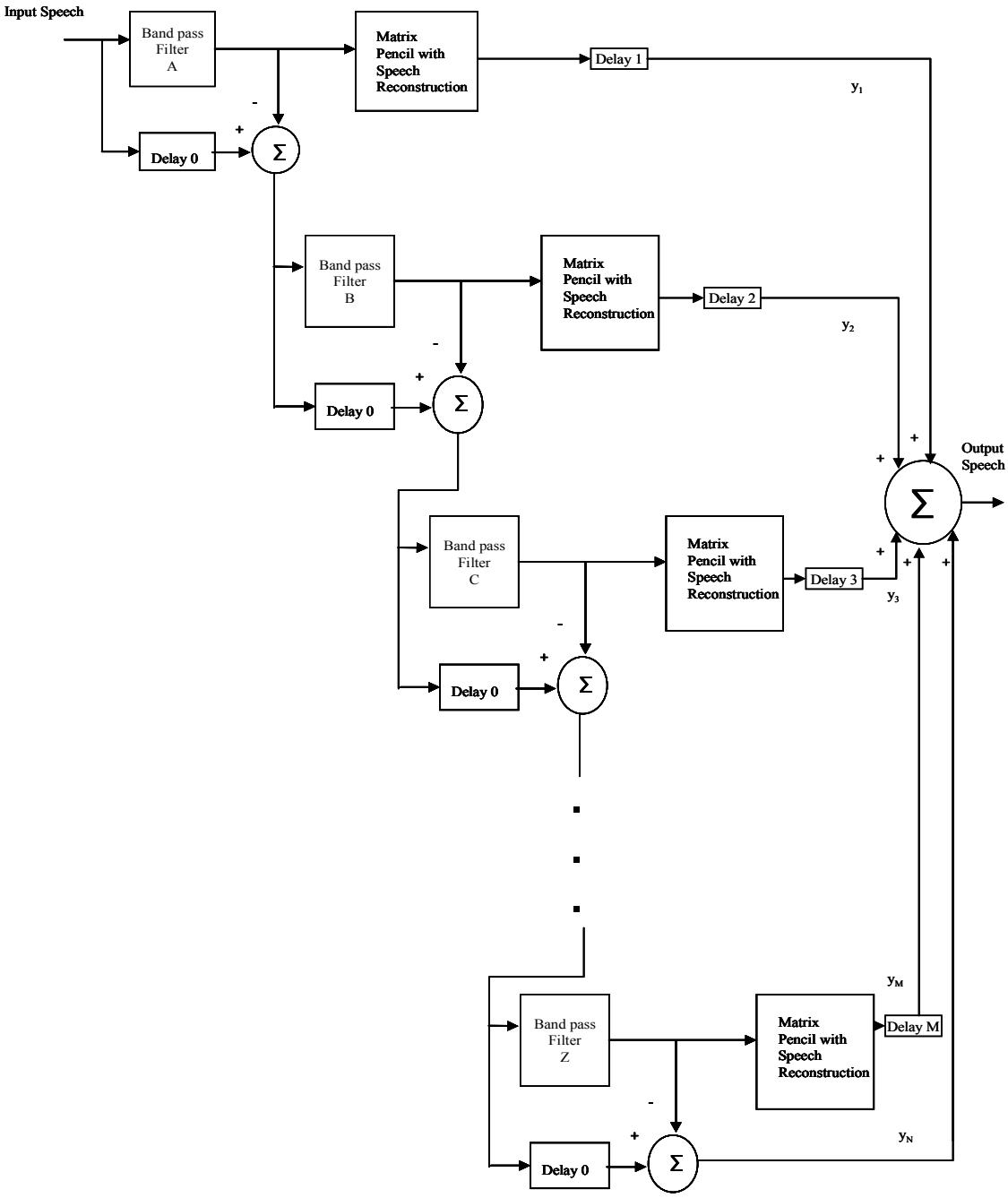

Figure 4.19: Block Diagram of the Cascade BFMP-TR.

4.3.4 Single Tone, FM Signal, AM Signal, and Multi-Tone Testing

In this research, two metrics were used to measure how well the interfering signal was removed from the contaminated speech. They were SID and RMSE. Each of these

metrics measures different qualities of the speech signal. SID obtains information from the human vocal tract. This test will determine how well the interferer is removed from the speech signal, without removing any critical vocal tract information. The vocal tract information resides at the low frequencies. Therefore, contaminated speech with a 400 Hz tone would be a good test to determine if a speech enhancement method preserves the vocal tract information. Another reason this frequency was selected is that the power supply frequency of some aircraft is at 400 Hz. On such an aircraft, it is possible that a speech communication channel would have interference from the power frequency.

The second metric used in this testing is the RMSE. This is the measure between the original audio signal, prior to contamination, and the enhanced speech signal, after an interferer removal technique is applied. This measures how well the interferer is removed while preserving the original speech. The smaller the RMSE value the closer the enhanced audio signal is to the original signal.

The SID algorithm automatically identifies an individual by their "voice print". It decides if the speaker is a specific person, or is among a group of persons. SID is text independent; it does not depend on the phrase the speaker speaks [43]. There are two main types of SID systems, close set and open set. A close set system is confined to a group of speakers, which were trained into the system. An open set SID is a much more complex system. It allows unknown speakers to be tested, without the system being trained on them beforehand. In this experiment an algorithm for a close set SID is used. The SID algorithm in this experiment uses a cepstral feature set to determine the results, and has been used quite extensively in this type of application [44]. The cepstral based SID algorithm is very sensitive to environmental changes, such as a tonal interference.

The removal of an interfering tone with a notch filter has proven to improve the SID results [45].

The RMSE measures the difference (error) between the two signals (original and enhanced) on a sample by sample basis. The difference is squared and integrated throughout the signal and divided by the number of samples N. Finally taking the square root completes the RMSE, as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(X_i - Y_i)^2}{N}}, \tag{4.1}$$

where X is the original signal and Y is the enhanced signal.

The data used in these experiments is called the TIMIT Acoustic-Phonetic Continuous Speech Corpus. The data used consist of 190 audio signals, 38 speakers, speaking 5 sentences each. The specific files that were used in this experiment are listed in appendix A. This corpus consists of text phrases, and is designed to provide speech data for acoustic-phonetic studies and for the development and evaluation of automatic speech recognition systems. These sentences are phonetically rich sentences, down sampled to 8 kHz [38].

4.3.4.1 SID and RMSE in the presences of a Stationary Interfering Tone

In the first experiment a stationary 400 Hz tone at 0 dB was inserted in the TIMIT data. Many different removal methods were tested. The first method included the notch filter at many different bandwidths with 100, 1000, and 5000 taps. The second and third methods were the BFMP-TS and BFMP_TR at different bandwidths for its band pass filter at a variety of different tap sizes. Figure 4.20 and 4.21 show the results for removal

of this tone using these methods. Figure 4.20 show the results with SID, and Figure 4.21 shows the results of the RMSE test. The baseline with speaker identification on original speech is 96.8%. This baseline is the SID score of the original data with no interference. The objective of this test is determine which method removes the 400 Hz tone the best, by scoring the closest to the baseline. The notch_xx nomenclature refers to a notch filter, with xx being the percent of the bandwidth at the respective frequency of the tone. Many different size notch filters were used to show that an optimum bandwidth can be compared to the other techniques. It can be seen in the results, that the optimum bandwidth maybe different for SID and RMSE tests. It can be observed that the BFMP-TR algorithm performs very well and close to the baseline, better than the other tone removal techniques. The different tap sizes and the size of the bandwidth of the pass band do not change the results. It is very robust to the bandwidth, since it works extremely well with a fixed ideal value. The BFMP-TS technique didn't fair as well as the BFMP-TR. The 15% BW notch filter, the observed optimum filter, was also tested with 100 and 1000 taps to show how the results degrade with this method. Observe that the BFMP-TR performs very well, consistent, and favorable when compared to the notch filter under the same parameters.

The RMSE results of the same data also showed that the BFMP-TR performed the best in the removal of this type of tone. As it is shown in Figure 4.21, the lowest RMSE scores come from this technique. Once again this technique is very robust to the size of the pass band, and to the number of taps used. The notch filter is not as robust, and does not perform as well as the BFMP-TS.

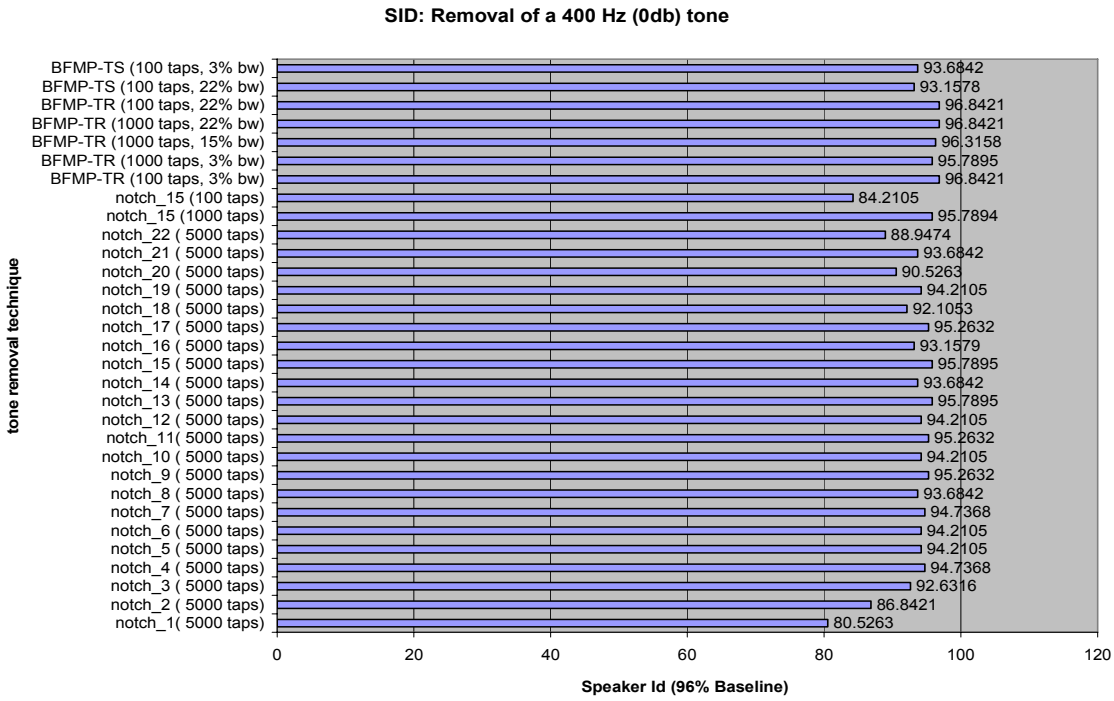

Figure 4.20: SID results after a 400 Hz tone was removed using multiple removal techniques.

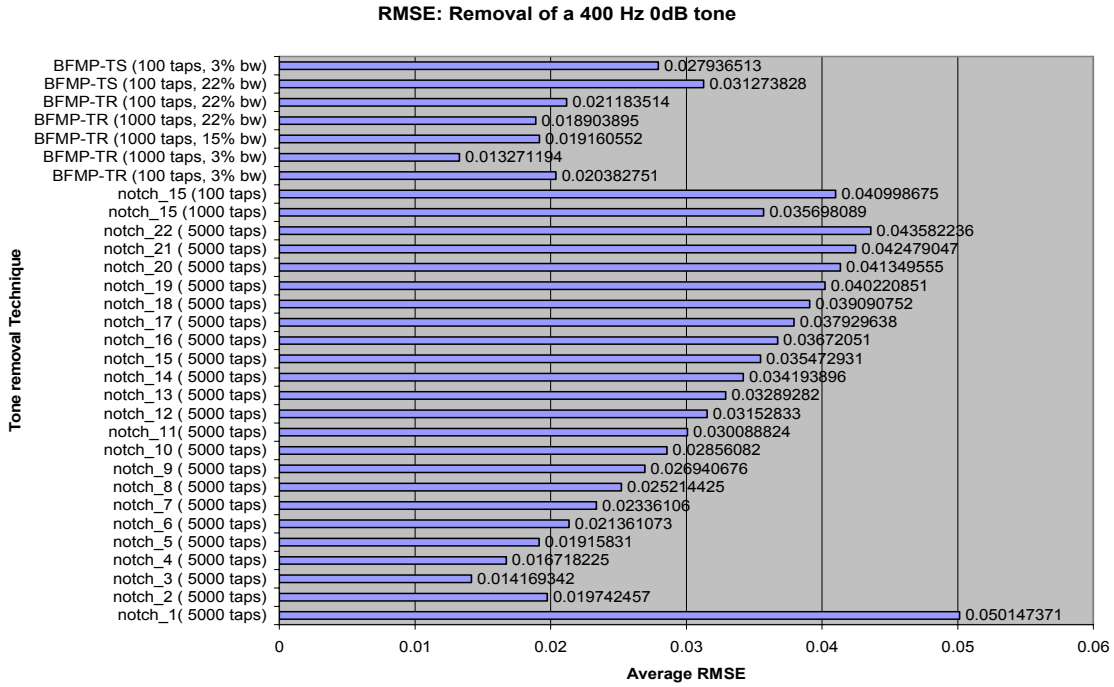

Figure 4.21: RMSE results after a 400 Hz tone was removed using multiple removal techniques.

4.3.4.2 SID in the presences of Multiple Amplitude Level Interfering Tone

In another speaker identification test, the amplitude of the interfering tone was varied with respect to the speech signal. The interfering tone was varied from -30 dB to 30 db in increments of 10 db. This test measures how effective and consistent the tone removal technique removes a tone that may have a large or small amplitude. The three techniques that were used are as follows: BFMP-TR, Triangular, and Blackman methods. The Triangular and the Blackman methods are a variation of a notch filter, using either a Triangular or a Blackman window respectively [8].

The results are shown in Figure 4.22 and 4.23. Figure 4.23 is the zoomed in version of Figure 4.22. The baseline score for correctly identifying the speakers in the original audio is 96.8%. Observe how poorly SID does on an audio signal that has a high interfering amplitude tone. This demonstrates the effect of an interfering tone on SID. All of the tone removal techniques do a fine job in removing the tone with respect to this test. Examining the results in Figure 4.23, the tone removal techniques tend to separate from each other with respect to consistency and performance. As it is shown, the Blackman technique does not perform well when removing a small amplitude tone. Likewise when the interferer tone has a large amplitude, the Triangular method performance decreases. Both are not consistent when removing tones that have different amplitude levels. The BFMP-TR performs the best out of all of the methods. It is very

consistent; it does not vary its results from one interferer level to the next. The performance of the BFMP-TR is excellent, with a speaker identification score of 96.3%.

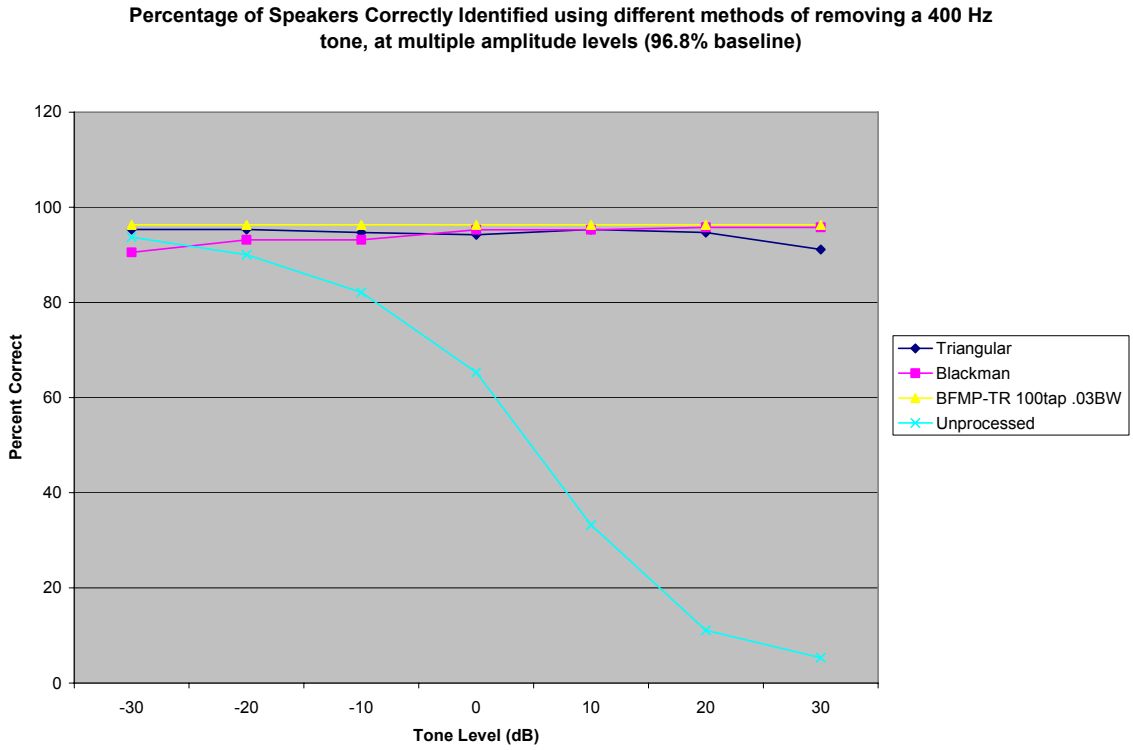

Figure 4.22: Speaker Identification on the removal of multiple tone amplitude levels.

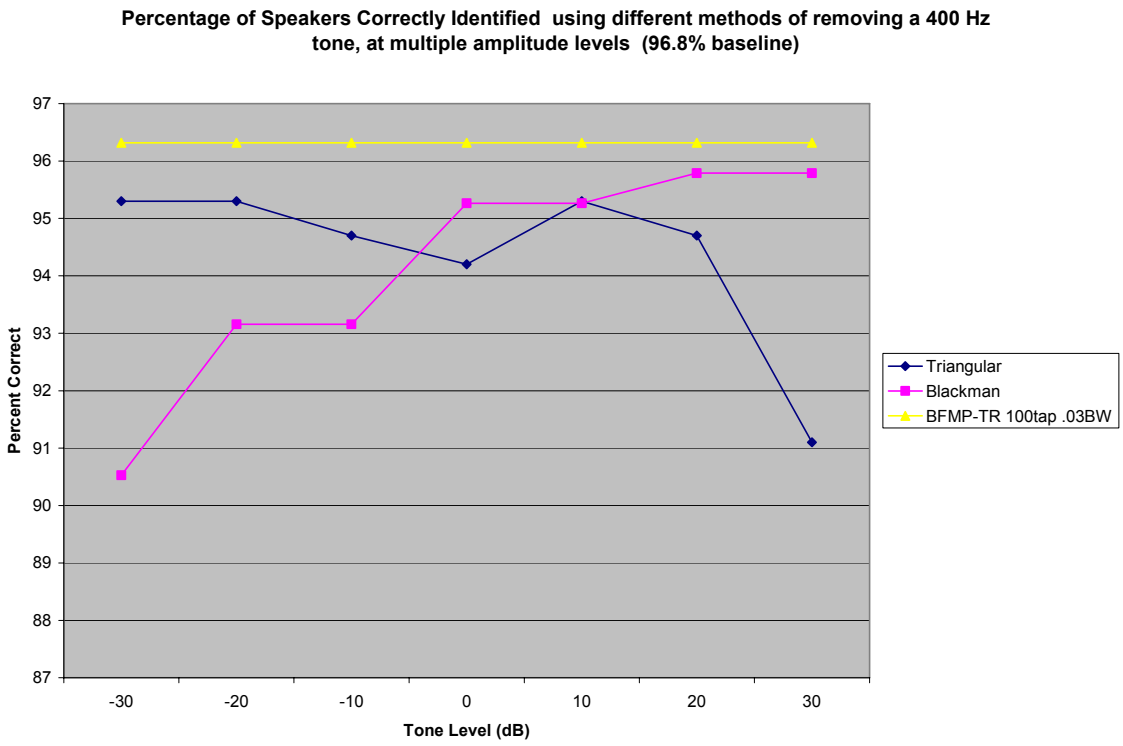

Figure 4.23: Zoom in version of Figure 4.22.

4.3.4.3 SID and RMSE in the presences of a FM Interfering Signal

The next type of test that was performed was the removal of a FM signal from the same data set. Three different interfering FM signals were removed by these different techniques. The difference between the three signals is the modulating index and the modulating frequency. The first interfering signal has a modulating index of 10 and the modulating frequency (f_m), of 2 Hz, giving a delta frequency (Δf) of 20 Hz. The second interfering signal has a modulating index of 10 and a modulating frequency of 1 Hz resulting in a Δf of 10 Hz. The third interfering signal has a modulating index of 5 and a modulating frequency of 1 Hz giving a Δf of 5 Hz. Each interfering signal was centered

around 400 Hz, mixed with the original signal and then removed using the methods discussed.

The results of the three FM data sets are shown in Figures 4.24 thru 4.29. For each FM signal interferer, the results were similar. Observing the SID results, in Figures 4.24, 4.26, and 4.28 shows that the BFMP-TR technique are equivalent or better than a notch filter, depending on the parameters of the notch filter. When additionally considering the group delay, the BFMP-TR technique is favorable over a notch filter. This is observed by comparing the notch_22 500 tap with the BFMP-TR whose delay and bandwidth size are the same. For the comparison to be similar the notch's group delay needs to be increased to 5000 taps and the bandwidth of notch filter is restricted to a small group of bandwidth sizes. Included in the BFMP-TR technique is the threshold, the value of which is the maximum Euclidean distance of zeroed poles. All poles within the Euclidean distance of the targeted pole are cancelled. Since this is a FM signal, for some frames, it may be necessary to cancel multiple poles, while at other frames it may be necessary to only cancel one pole. Therefore an Euclidean distance threshold was used to achieve this procedure. Note that the amplitude of the interfering signal was large enough to do detection on a frame by frame basis.

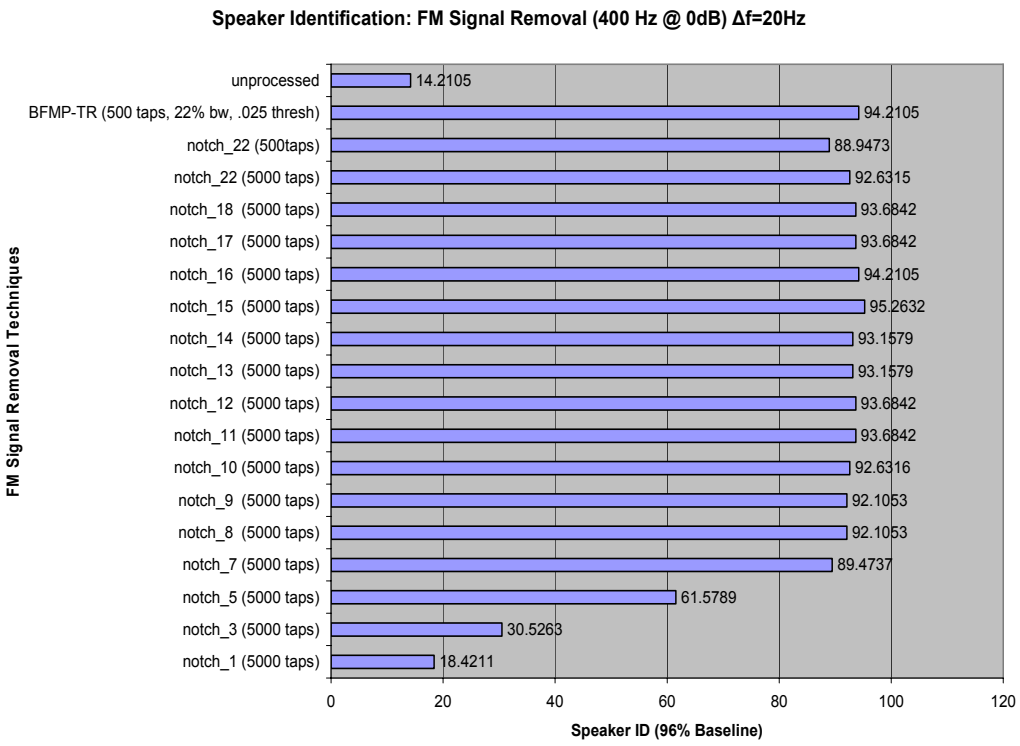

Figure 4.24: SID results after a 400 Hz FM signal (Δf=20Hz, f_m=2Hz) was removed using multiple removal techniques.

The RMSE test shows similar results to the SID results. In these results the BFMP-TR performs very well and better than the notch filter. The BFMP-TR method has the smallest RMSE and the smallest group delay for each of the FM signal interferers. Another conclusion is that the notch filter, for a particular delay and bandwidth size, may do well for SID but it may not do well for the RMSE test. As for the BFMP-TR it does well for both tests. Therefore for a FM signal interferer, the BFMP-TR is favorable for removing FM signal interference from a speech signal.

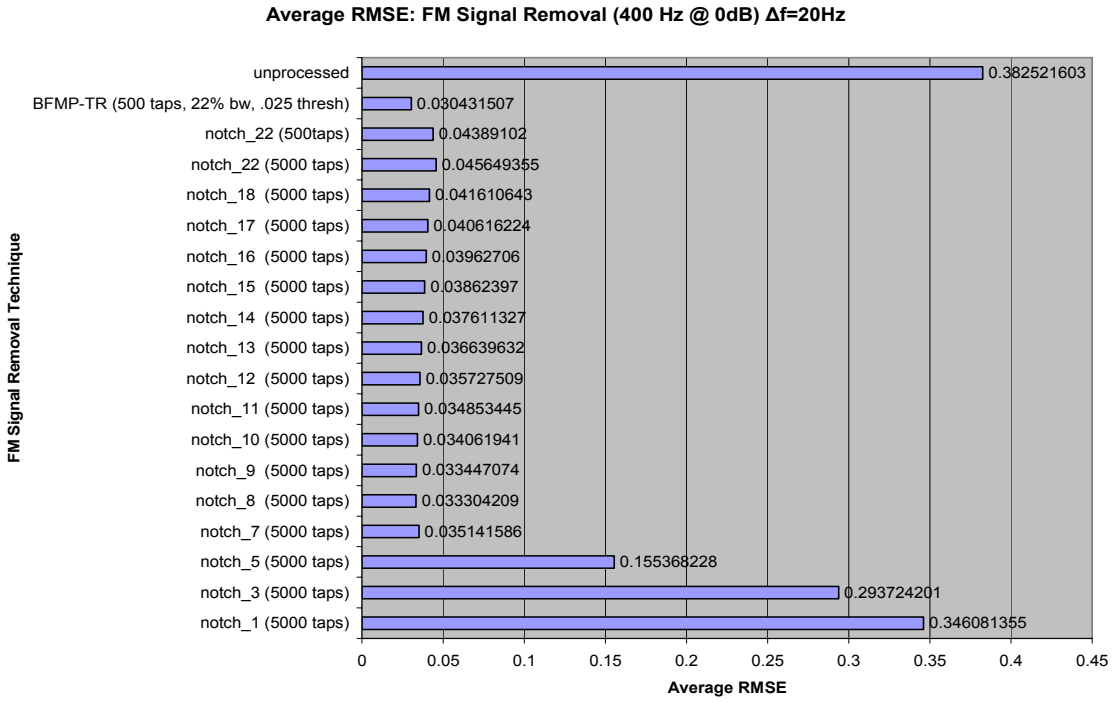

Figure 4.25: RMSE results after a 400 Hz FM signal ($\Delta f = 20Hz$, $f_m=2Hz$) was removed using multiple removal techniques.

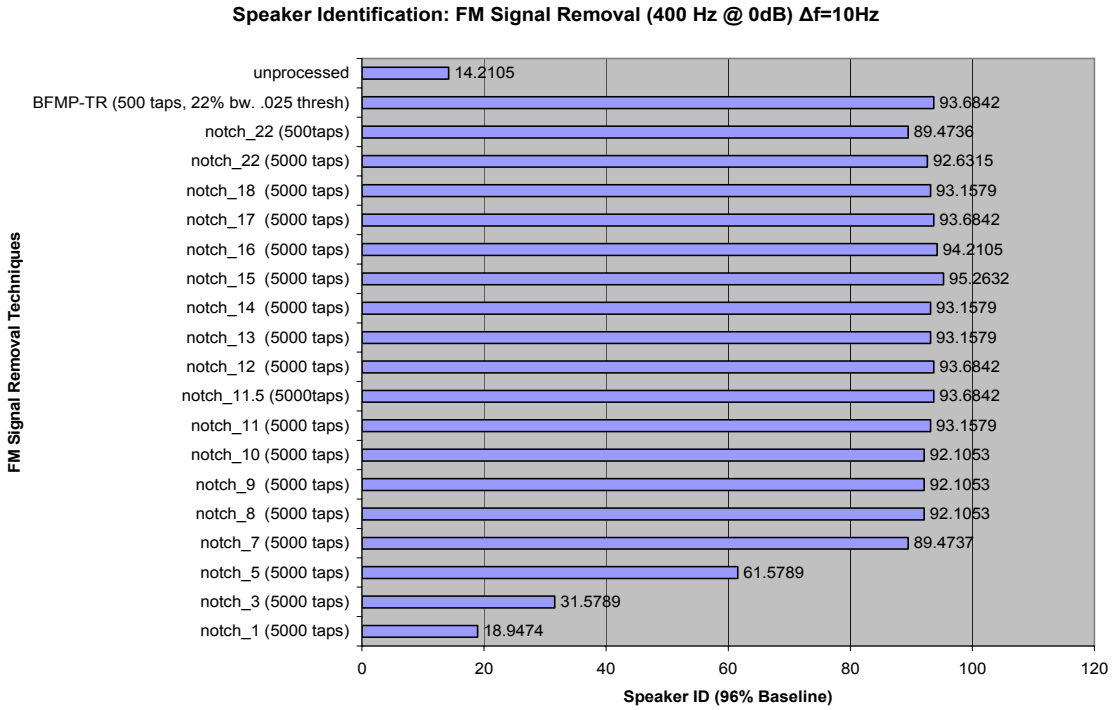

Figure 4.26: SID results after a 400 Hz FM signal ($\Delta f = 10Hz$, $f_m=1Hz$) was removed using multiple removal techniques.

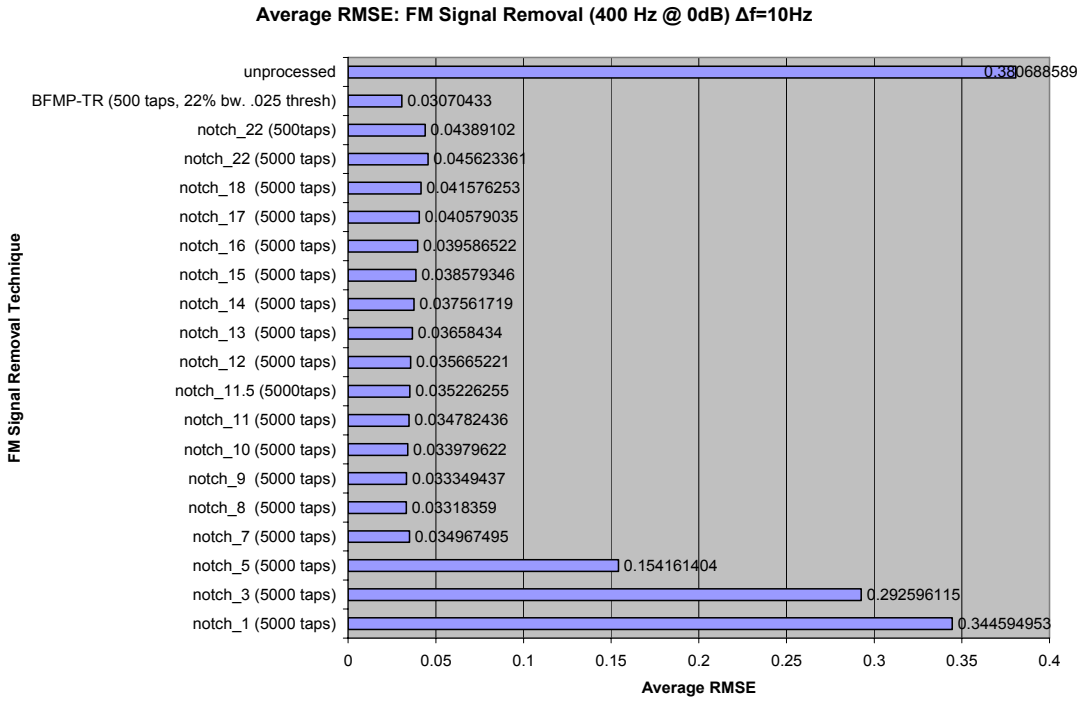

Figure 4.27: RMSE results after a 400 Hz FM signal ($\Delta f = 10$Hz, f_m=1Hz) was removed using multiple removal techniques.

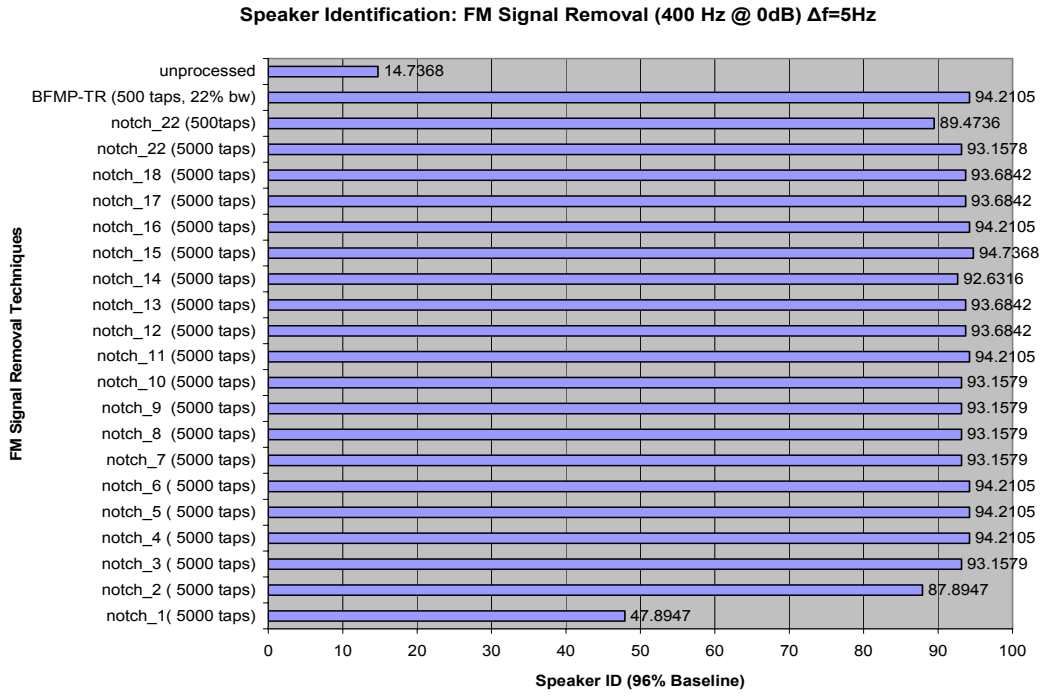

Figure 4.28: SID results after a 400 Hz FM tone ($\Delta f = 5$Hz, f_m=1Hz) was removed using multiple removal techniques.

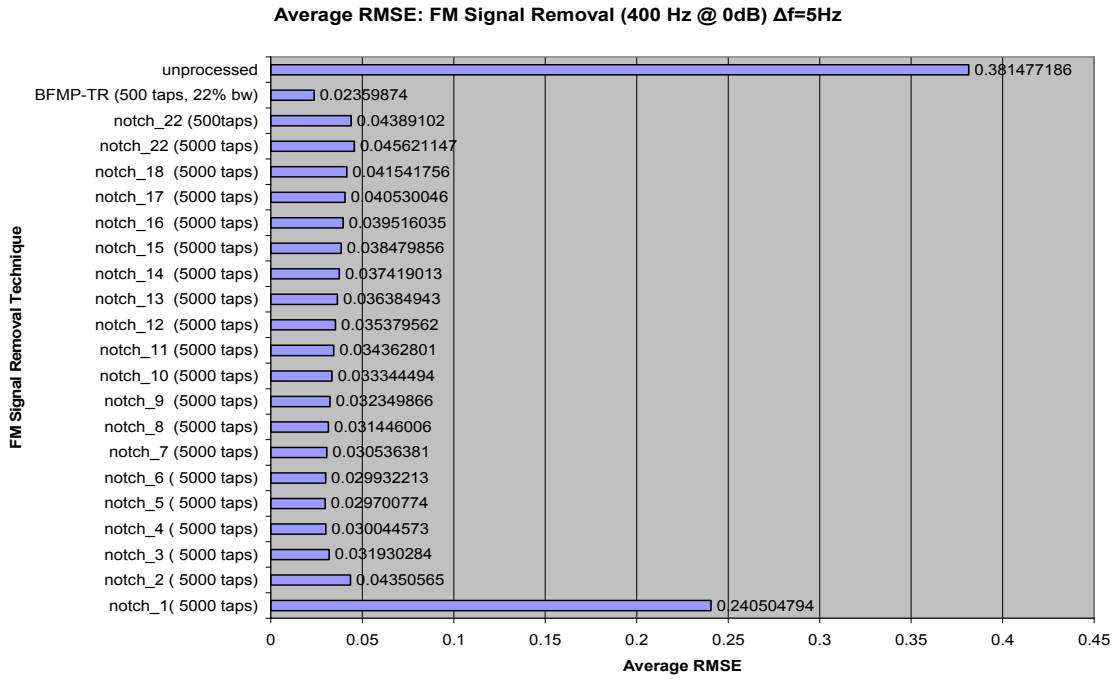

Figure 4.29: RMSE results after a 400 Hz FM signal ($\Delta f = 5Hz$, $f_m=1Hz$) was removed using multiple removal techniques.

4.3.4.4 SID and RMSE in the presences of an AM interfering signal

The third type of interfering signal that was tested was an AM signal. When a sinusoidally modulated AM signal is mixed with an audio signal, the resulting signal has three interfering tones. There is one at the carrier, a second and third tone above and below the carrier plus and minus the modulating frequency respectively. Since there are multiple tones, removing this type of signal is a difficult problem. A typical notch filter would have to have a large bandwidth, especially if the modulating frequency is large. A speech signal that has had an AM interfering signal tone removed, would also have more speech information removed.

In the AM signal removal test, a 400 Hz 0 dB AM tone with a modulating frequency of 25 Hz and a modulation index of .5, was added to the audio signals. It was

then removed with the techniques previously discussed. The results are shown in Figure 4.30 and 4.31. As can be seen, the BFMP-TR is again the favorable technique. As in the other SID tests, the BFMP-TR has the advantage in both the identification results and the group delay. The identification results for the BFMP-TR technique is equivalent to the baseline. As for the performance of the notch filters, they all fall below the baseline. In the RMSE test the results are similar to both the FM interfering signal tests and the 400 Hz stationary interfering tone test. The BFMP-TR technique has better or similar performance relative to the notch filters, but when the group delay is considered, this BFMP-TR technique is more favorable. The group delay is nearly ten times less than the delay for the notch filter.

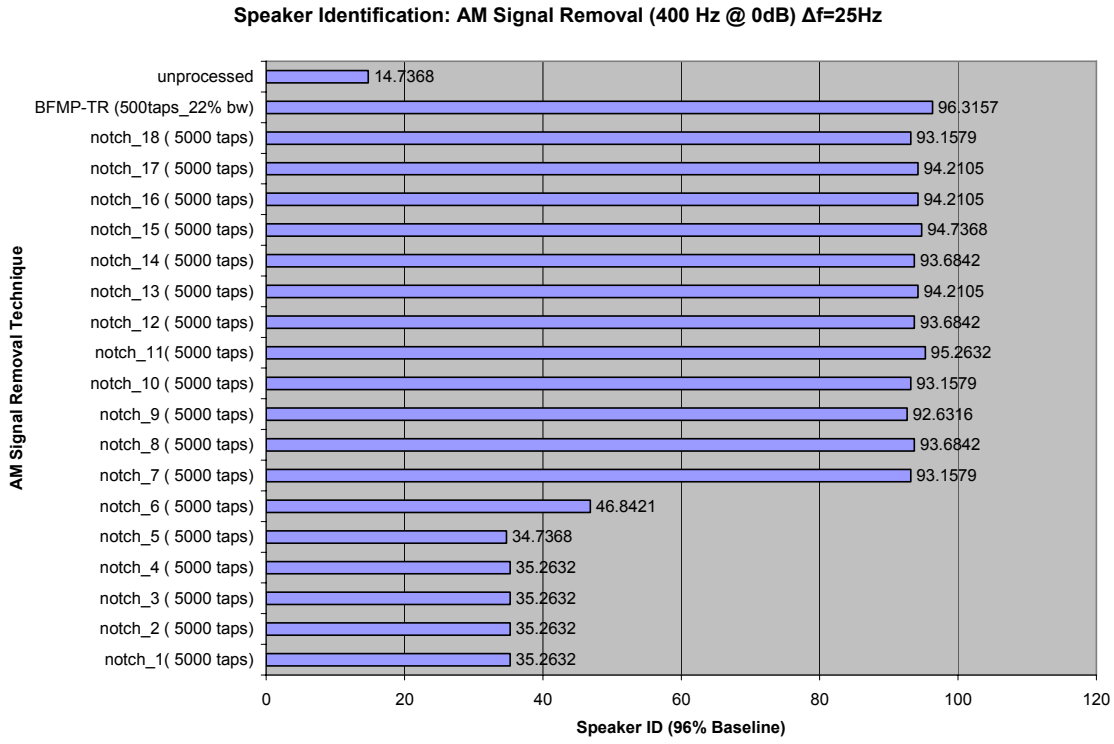

Figure 4.30: Speaker Id results after a 400 Hz AM tone (f_m = 25 Hz, modulation index = .5) was removed using multiple removal techniques.

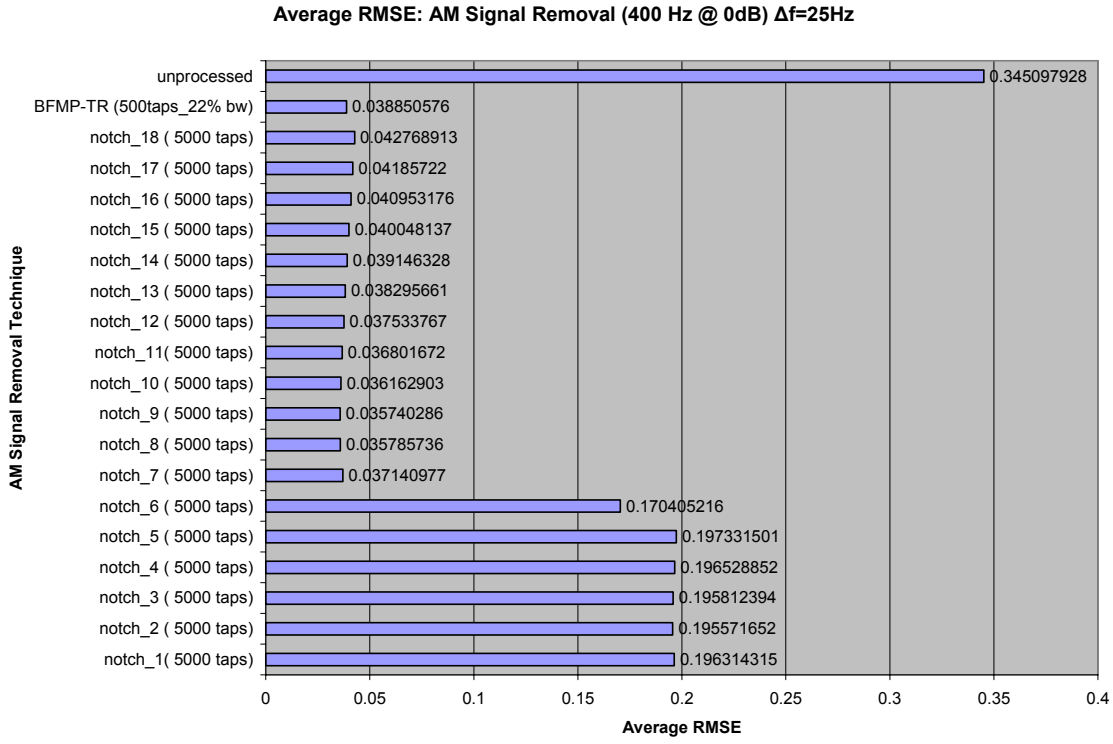

Figure 4.31: RMSE results after a 400 Hz tone AM tone (f_m = 25 Hz, modulation index = .5) was removed using multiple removal techniques.

When increasing the modulating frequency larger than 25 Hz, it becomes necessary to use multiple filters to remove these AM tones. For a notch filter, it becomes necessary to use three filters since the spacing of the tones are farther apart. This avoids removing too much speech information. This problem also occurs for the BFMP-TR. The solution to this problem is to implement a cascade BFMP-TR system previously discussed. This will pass multiple bands to reduce the amount of speech information that is lost, while removing all interfering tones as in the following example.

4.3.4.5 SID and RMSE in the presences of a 60 Hz Interfering Tone and its Odd Harmonics

This type of test signal consists of a 60Hz tone plus its first 4 odd harmonics at 180Hz, 300 Hz, 420 Hz, and 540Hz. Each of the five tones is at equal amplitude at 0dB relative to the speech signal. Two methods were used in this experiment, the notch filter and the cascade BFMP-TR. Like the other experiments, the notch filter's BW was varied between 1% and 8% of the interfering tone location; this was to determine the optimum BW. For the SID test, Figure 4.36, the BW that was favorable is 7%, which gave a SID result of 92.6%. The optimum BW for the RMSE test, Figure 4.37 was at 3%. The size of the BW was not consistent between the two different experiments. The results for the BFMP-TR were more favorable and more consistent than the notch filter, and also resulted in a smaller group delay. The BFMP-TR threshold is the percentage of the BW for each of the band pass filters. Observe when the threshold is set to a fixed number, the results decrease. The BW should increase for higher frequency tones, and be made smaller for lower frequency tones. This will assure that each tone is removed efficiently. Observe the spectrograms of the original signal, contaminated signal, enhanced signal via the notch filter, and enhanced signal via the BFMP-TR method. In Figure 4.33, it can be seen how tainted the contaminated signal is, and how well the two methods perform as seen in Figures 4.34 and 4.35. Compare the BFMP-TR spectrogram with the original spectrogram. The BFMP-TR removes the tones, while retaining most speech information. The notch filter leaves holes in the spectrum, losing valuable speech information.

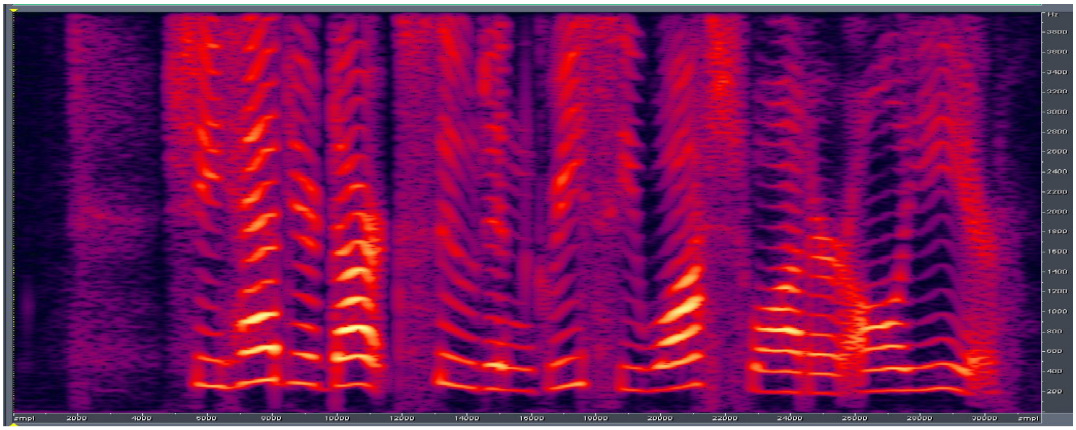

Figure 4.32: Spectrogram of the original signal
Note: x-axis is in samples (0-32000), y-axis is in frequency (0-4kHz).

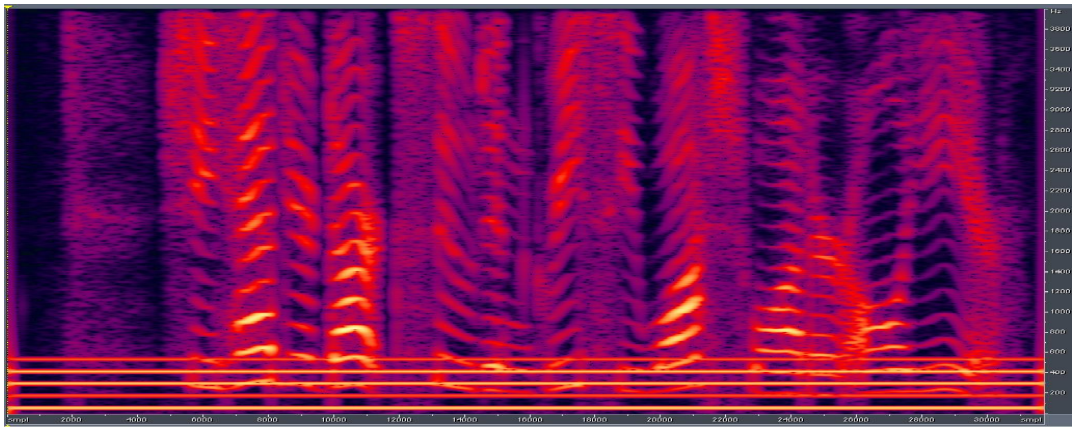

Figure 4.33: Spectrogram of the contaminated signal
Note: x-axis is in samples (0-32000), y-axis is in frequency (0-4kHz).

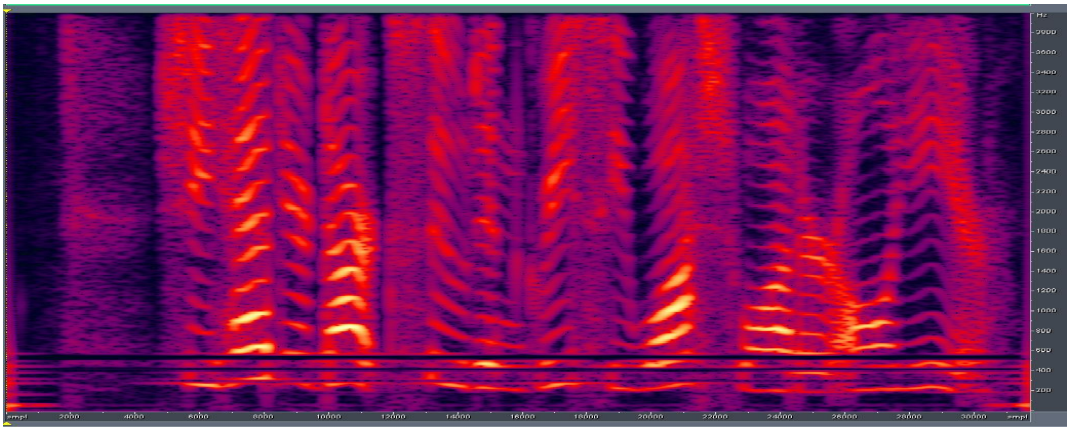

Figure 4.34: Spectrogram of the enhanced signal using the Notch Filter @ 7%
Note: x-axis is in samples (0-32000), y-axis is in frequency (0-4kHz).

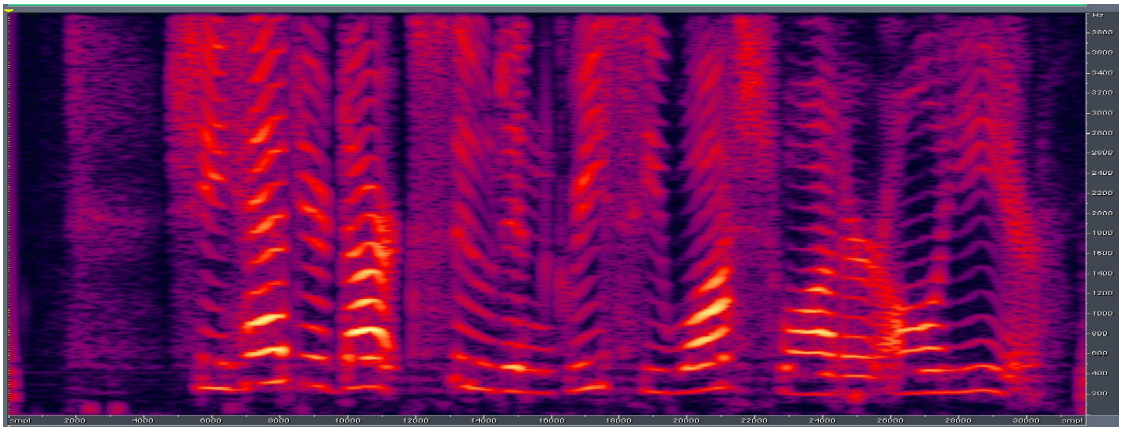

Figure 4.35: Spectrogram of the enhanced signal using the BFMP-TR @ 5%
Note: x-axis is in samples (0-32000), y-axis is in frequency (0-4kHz).

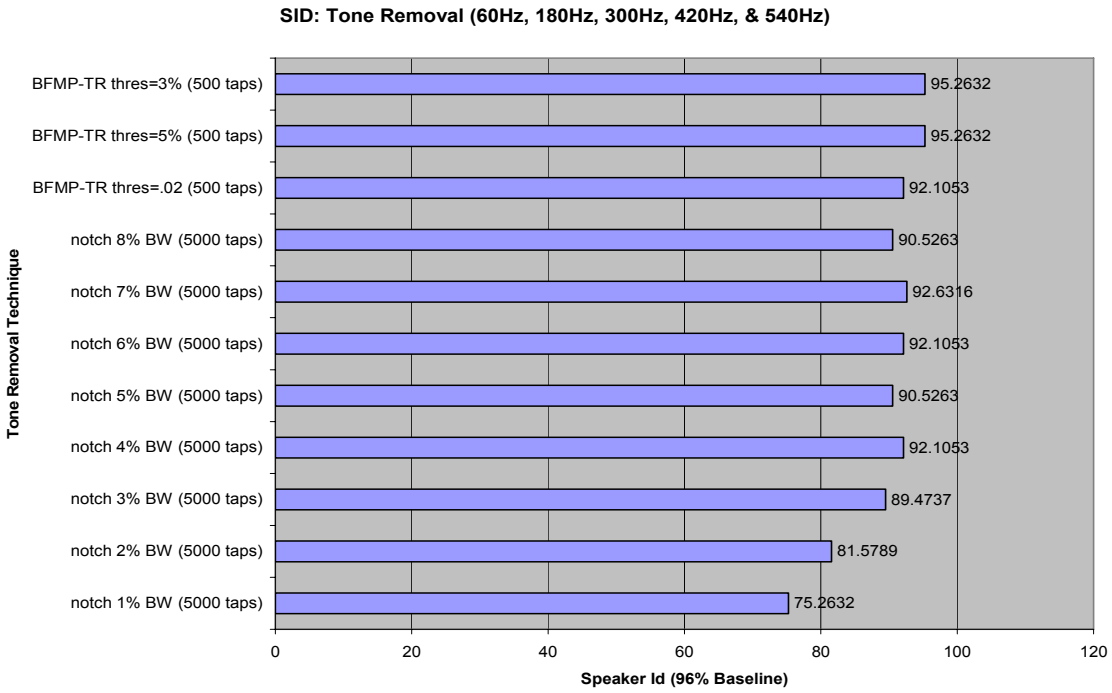

Figure 4.36: SID results after removing multiple harmonic tones (Baseline 96.8%).

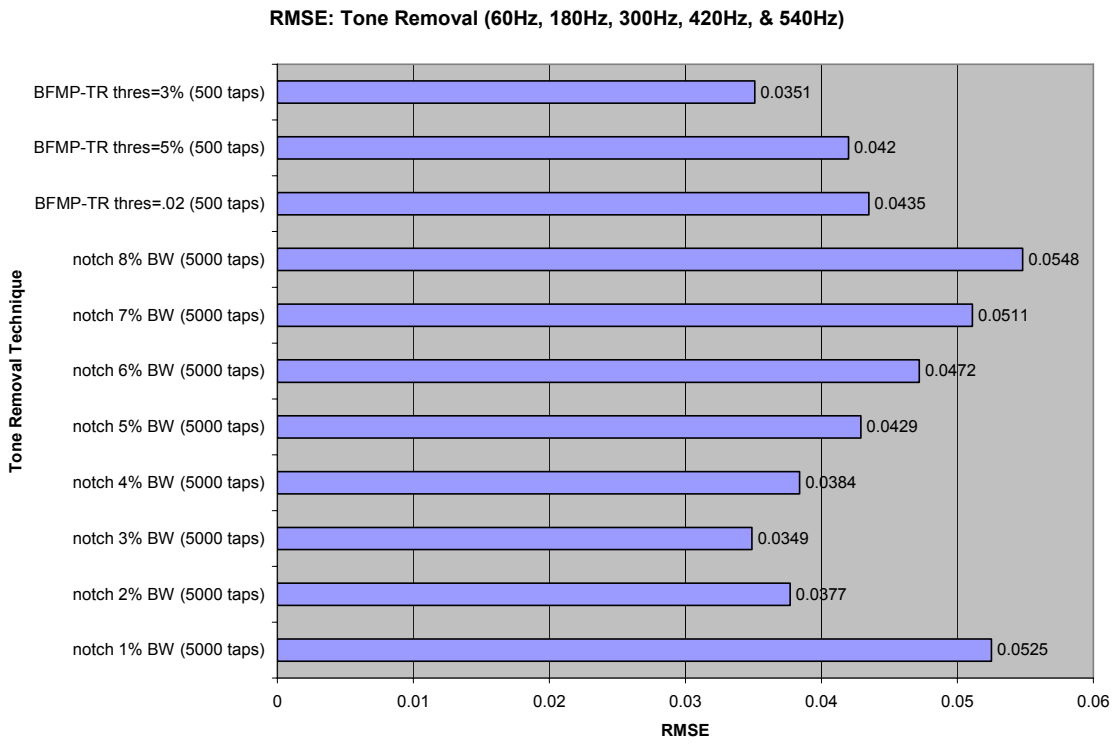

Figure 4.37: RMSE results after removing multiple harmonic tones.

4.4 Conclusion

In this chapter, a MP based filter was developed, called the band focus MP with temporal reconstruction (BFMP-TR). It is a filter that removes unwanted tonal interferes from a speech signal, while removing little speech information. This high resolution technique allows the tone to be more readily separated from the speech. The BFMP-TR algorithm was compared to a typical notch filter. The BFMP-TR group delay is much smaller than the notch filter's delay with equivalent or better results. This would allow the BFMP-TR algorithm to be used in real time systems.

The results show that the BFMP-TR algorithm works well in removing many different types of interferers. These interferers include stationary tones, FM signals, and

AM signals. It was shown that a cascade based BFMP-TR system can be implemented to remove multiple tones, with excellent results. This cascade system could also be implemented with an AM interferer, whose modulating frequency is larger than 25Hz.

The BFMP-TR can be applied to many different types of speech enhancement problems. There are other types of interferers that cause problems to speech processing applications. These problems can potentially be addressed with the MP algorithm. Wideband noise, particular white noise, is a problem in military applications. Crowd noise as in a cocktail party, is a problem with individuals wearing hearing aides. Removing these noises in real time is a challenge. The MP based filter gives these problems a potential for a highly robust solution.

Chapter 5

Pitch Tracking using the Generalized Harmonicity Indicator

5.1 Introduction

For many audio applications, a process is required to obtain an accurate estimate of the fundamental and harmonics of periodic sections of the audio signal. More generally, any digital version of a periodic signal can potentially have an associated fundamental frequency component, along with harmonics which are frequency components located at integer multiples of the fundamental. In this description, the focus will be on audio applications and speech applications in particular, without loss of generality to applications outside the speech and audio domains.

For speech, tracking and assessment of fundamental and harmonic frequencies can be a key step in accomplishing such tasks as automated speaker identification, speech data compression, pitch alteration and natural sounding time compressions and expansions [46]. Linguists and speech therapists also use such tracking and assessment for prosodic analyses and training [47].

Various methods of fundamental and harmonic frequency tracking have been proposed and developed, but most have been based on other low resolution techniques such as FFT and cepstral analyses [46]. This is as opposed to using super-resolution

frequency estimation as provided by the MP technique. The prior art in the area of super-resolution speech fundamental determination consists of the "super resolution pitch determinator" (SRPD) [48] and the "enhanced super resolution pitch determinator" (eSRPD) [49] methods. Because these prior methods do not explicitly process a spectral representation or decomposition of the input audio signal, they are not considered to be in the same class as the MP GHI. However, the SRPD and the eSRPD do provide a baseline for comparisons when assessing the performance of an MP based GHI and will therefore be referred to in the context of performance. Another method worth mentioning is an autocorrelation method described in [50]. In this paper the author emphasizes the comparison of the gross error, without comparing the deviation mean or the standard deviation, such as in [49] and used here.

The purpose of the GHI is to determine, assess and track the fundamental and harmonic frequencies of consecutive time segments of a signal.

5.2 Pre Processing

As a pre-processing step to the GHI process, and similar to prior sections, the signal to be analyzed is first divided into consecutive overlapping or non-overlapping frames. Frame lengths and overlap percentages are typically chosen to be consistent with the stationary properties of the signal to be analyzed. In particular, multiple periods should be present in the segment, but the number of periods should not be arbitrarily large otherwise the fundamental and harmonic values may deviate excessively. Also, choosing too many periods can cause the computational complexity of super-resolution techniques to become prohibitive. Therefore, for this experiment it was determined that a

frame size for a male speaker is near optimal at 25.6 ms, with a 50% overlap, corresponding to a 12.8 ms time steps for the beginning of each frame. For a female speaker the frame size is optimal at 12.8 ms, with a time step of 6.4 ms. Since a male speaker has a lower pitch period than a female speaker, it was necessary to use a larger frame size to capture the whole pitch period.

For each segment, a second pre-processing step is the calculation of the super-resolution representation of the segment, as provided by signal decompositions as in the MP technique. As stated in the previous sections the MP technique is particularly effective at determining the frequency content of the signal, and includes frequency, decay rates, initial phases, and initial amplitudes in the decomposition. For each frame, 28 poles in the forward mode were determined to work well.

In a third and final pre-processing step, available decay and initial amplitude values are used to prune the original list of frequencies that the super-resolution process provides from the segment being decomposed. Frequencies that are too close to each other within the frequency resolution of the technique are eliminated. Likewise, frequency values that are not tone-like due to non-trivial decay (or growth) values are also eliminated. Therefore, poles with an absolute value of the decay coefficient less than .01 were also eliminated. Any zero valued frequencies that may result are also eliminated. The final pruning is the elimination of frequency values associated with trivial initial amplitudes relative to the number of bits of precision in the representation of the digitized signal. Therefore, poles with an amplitude less than $1/2^{16}$ were eliminated. The result is a list of frequency values $\vec{L}$, which serves as input to the GHI process.

Reference is made to Figure 5.1 for a description of the GHI process which consists of the sequence of steps that follow.

5.3 Voiced/ Unvoiced Detection

In speech analysis, the voiced/unvoiced decision is often performed in conjunction with pitch analysis. The linking of the voiced/unvoiced decision to pitch analysis not only results in unnecessary complexity, but makes it difficult to classify short speech segments which are less than a few pitch periods in duration [51]. Therefore, it is better to classify a speech frame as voiced speech, or unvoiced speech, separately from estimating the pitch.

A voiced frame consists of a fundamental frequency with several related harmonics. An unvoiced frame does not have this property. A voiced/unvoiced detector determines if a speech frame is voiced or unvoiced. A voiced/unvoiced detector can rely on a few parameters to accomplish its goal. These parameters are energy, zero crossing, or prediction gain. Relying on one parameter limits the robustness of the voicing detector. Whereas, increasing the number of parameters will increase its reliability [37].

In this experiment, the goal is to use a similar voiced detector that was used in [49]. This would make the comparison between the two pitch estimation algorithms fair. In [49] the author uses an energy based detector that uses a threshold. The singular values from the SVD are related to signal energy; therefore, they can be used by a voiced detector. Also, since the SVD is already used in the calculations of the MP algorithm, there is little or no processing overhead to use it as a voiced detector. To determine a voiced frame from an unvoiced frame, a threshold is needed. The maximum singular

value of the frame is compared to the threshold. The frame is classified as an unvoiced frame if the threshold is larger than the maximum value. Otherwise, it would be classified as a voiced frame. This threshold is varied until the voice in error is the same as in [49]. This provides for the comparison of the pitch results in a fair way.

5.4 Pitch Estimation Algorithm

The GHI process is shown in Figure 5.1. After the MP algorithm generates a set of frequencies for a given frame, preprocessing eliminates some frequencies as previously described. The result is a list of frequency values $\vec{L}$, which serves as input to the GHI process. The n elements of the nx1 list vector $\vec{L}$ are ordered in the Frequency Sorter, for example in ascending order, to form the ordered frequency list vector, $\vec{F}$. The nx1 vector $\vec{F}$ is then input to the Column Duplicator, which forms the nxn matrix $\mathbf{F}$ by replicating $\vec{F}$ for each column of $\mathbf{F}$. Thus $\mathbf{F} = \vec{F}\vec{1}^T$, where $\vec{1}^T$ is a 1xn dimension row vector, the elements of which are all 1. The frequency matrix $\mathbf{F}$ is then input to the Candidate Generator, where the nxn matrix of candidate fundamentals, $\mathbf{D}$ is formed as $\mathbf{D} = \mathbf{F} - \mathbf{F}^T$. When ascending ordering is used for $\vec{F}$, the matrix $\mathbf{D}$ can be represented as the sum of an upper triangular matrix and a lower triangular matrix, and will have diagonal elements that are each zero. Thus the elements below the diagonal for the described ascending ordering will be the frequency differences which can be used to determine the fundamental and harmonics in subsequent steps.

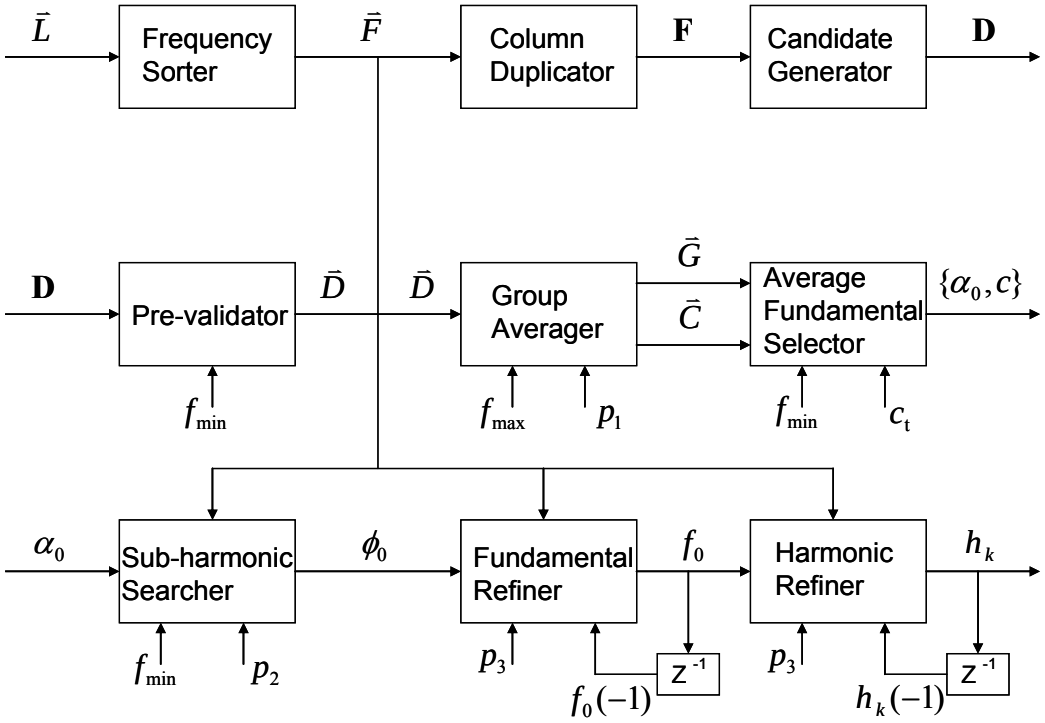

Figure 5.1: The Generalized Harmonicity Indicator.

The matrix **D** is input to the Pre-validator which forms a vector $\vec{D}$ whose elements are chosen from the positive elements of **D** that are greater than some minimum value, $f_{\min} > 0$. The elements of the mx1 vector $\vec{D}$ are arranged in ascending order and will result in $m \le 0.5n^2 - 0.5n$. The pre-validated candidate fundamental list, $\vec{D}$, is then input to the Group Averager, which produces both a vector of averaged groupings of fundamentals, $\vec{G}$, and an associated count vector, $\vec{C}$. To generate the groupings, $\vec{G}$, group boundaries are formed by inspecting the elements of the candidate fundamental list, $\vec{D}$. Starting with the second element of $\vec{D}$, a difference is formed between each current element and the previous element in the vector. If this difference is less than a fraction p_1 times the current element, then the element is grouped with the prior element.

Otherwise, a new group is started with the current element. The parameter p_1 is typically chosen to be 0.1 (10 percent). Because elements are in ascending order, each group represents a distinct positive change in candidate fundamentals. For each defined group, the number of elements in each group are used as the elements of the count vector, $\vec{C}$. Using these counts, groups of candidate fundamentals are averaged to form the corresponding elements of the vector $\vec{G}$. Averages greater than the parameter $f_{\max}$ are not allowed, and likewise the corresponding elements of the count vector $\vec{C}$ are eliminated. The group average vector, $\vec{G}$, and the count vector, $\vec{C}$, are both input to the Average Fundamental Selector. If after such processing there are no elements in $\vec{G}$, then it is arbitrarily assigned a single element equal to $f_{\min}$, and the count vector $\vec{C}$ is assigned a corresponding single element equal to a count threshold, c_t. For example, the count threshold for the speech pitch estimation application was set to 3. From the group average vector, $\vec{G}$, a subset of elements is chosen which correspond to the largest elements of the count vector, $\vec{C}$, greater than or equal to the count threshold c_t. For the speech pitch estimation example application, the elements corresponding to the 3 largest counts are used. The initial fundamental estimate, α_0, is chosen as the minimum of the group averages from the subset. The count, c, is chosen as the largest count. Thus the Average Fundamental Selector is biased away from simply using the largest group average. This results in an enhanced selection process that allows for the possibility that a valid fundamental is not the one associated with the largest count.

The scalar value initial fundamental, α_0, and the associated count, c, are input to the Sub-harmonic Searcher. The Sub-harmonic Searcher forms the nx1 sub-harmonic

candidate vector as $\vec{S} = \vec{F} - 0.5\alpha_0 \vec{1}$ and uses this vector to determine whether or not α_0 should be reduced by a factor of 0.5. Reduction is performed if $0.5\alpha_0$ is greater than $f_{\min}$ while at the same time, the minimum of absolute values of the elements of $\vec{S}$ is less than $0.5 p_2 \alpha_0$. Here, p_2 is a fractional parameter that restricts the search space. A typical value for this parameter is 0.1 (10 percent). The resulting output of the Sub-harmonic searcher is designated as ϕ_0, and represents the fundamental estimate prior to optional refinement processes.

The pre-refined fundamental estimate, ϕ_0, is input to the Fundamental Refiner. A pair of nx1 error vectors are formed as $\vec{E}_{-1} = \vec{F} - f_0(-1) \cdot \vec{1}$ and $\vec{E} = \vec{F} - \phi_0 \vec{1}$. Here, $f_0(-1)$ is the refined fundamental estimate from the previous signal segment, and $\vec{F}$ is the ordered list vector from the output of the Frequency Sorter. Thus the z^{-1} block represents a unit segment delay. A scalar, $x = p_3 f_0(-1)$, is also calculated and is used to restrain the refinement process. Typical values for the fractional parameter p_3 is also 0.1 (10 percent). A comparison is made to determine if the minimum of the absolute values of the elements of $\vec{E}$ is less than the minimum of the absolute values of the elements of $\vec{E}_{-1}$ and is also less than *x*. If so, f_0 is the element of $\vec{F}$ associated with the minimum of the absolute values of the elements of $\vec{E}$. If both of these conditions are not met, then $f_0 = \phi_0$ (no refinement is made).

The output of the Fundamental Refiner, f_0, is input to the final optional step, the Harmonic Refiner. This step is identical in form to the Fundamental Refiner, and is repeated for all harmonic frequencies of interest. For example a harmonic is formed as

the product $\phi_k = kf_0$, where the integer *k* is greater than 1. A pair of nx1 error vectors are formed as $\vec{E}_{-1} = \vec{F} - h_k(-1) \cdot \vec{1}$ and $\vec{E} = \vec{F} - \phi_k \vec{1}$. Here, $h_k(-1)$ is the refined harmonic estimate from the previous signal segment, and $\vec{F}$ is the ordered list vector from the output of the Frequency Sorter. A scalar, $x = p_3 h_h(-1)$, is also calculated and is used to restrain the refinement process. Typical values for the fractional parameter p_3 is also 0.1 (10 percent). A comparison is made to determine if the minimum of the absolute values of the elements of $\vec{E}$ is less than the minimum of the absolute values of the elements of $\vec{E}_{-1}$ and is also less than *x*. If so, h_k is the element of $\vec{F}$ associated with the minimum of the absolute values of the elements of $\vec{E}$. If both of these conditions are not met, then $h_k = \phi_k$ (no refinement is made).

5.5 Results

Shown in Table 5.1 are the performances results for the MP GHI process for the application of speech pitch estimation which in the present context refers to fundamental frequency estimation. The top half of Table 5.1 refers to results from male speech and the bottom half refers to female speech. The speech database that was used, is described in [49], and was downloaded from the author's website [52]. This database consists of a female and male speaker each speaking 50 English sentences sampled at 20 kHz. This database includes the recording of laryngeal frequency for each speech file in the database, which acts as the ground truth for fundamental estimation. This laryngeal data was created by placing a sensor on the subject's throat, while the speech data was collected. As previously described, a

special property of speech is the fact that each segment of an utterance can be classified as either voiced or unvoiced. As implied, the voiced segments of the speech are segments that contain fundamental and harmonic frequency content, whereas unvoiced segments are either silence or fricatives and plosives. These latter segments contain either weak or no fundamentals and harmonics.

The ground truth given in this database consisted of the voiced frames time locations and their respective fundamental frequency. In this experiment, the start of each frame was slightly different than the ground truth time marks. Therefore, there was a need to adjust the ground truth to correspond to the beginning of each frame. The ground truth data fundamental frequency was interpolated for each sample. Once the data was interpolated, an interpolated fundamental frequency was determined by locating the beginning time of each frame. This allowed for one to adjust the frame size of the experiment and determine which frame size was optimal.

Table 5.1: Fundamental estimation evaluation for male speech (top) and female speech (bottom).

Method	**Unvoiced in error (%)**	**Voiced in error (%)**	**Gross high errors (%)**	**Gross low errors (%)**	**Absolute deviation (Hz) mean**	**Absolute deviation (Hz) p.s.d.**
SRPD	4.05	15.78	0.62	2.01	1.78	2.46
eSRPD	4.63	12.07	0.90	0.56	1.40	1.74
MP GHI	2.88	12.08	0.96	1.07	1.67	2.16
SRPD	2.35	12.16	0.39	5.56	4.14	5.51
eSRPD	2.73	9.13	0.43	0.23	4.17	5.13
MP GHI	0.89	9.06	1.09	0.22	3.02	3.91

To properly take into account the voiced/unvoiced classification process, Table 5.1 and 5.2 includes the percentage of voiced segments in error (voiced classified as unvoiced) and the percentage of unvoiced segments in error (unvoiced

classified as voice). This is necessary for a fair comparison because misclassifying voiced segments can affect important performance metrics, such as the absolute deviation mean and population standard deviation (p.s.d.). For example, a higher voiced in error percentage will cause the mean and p.s.d metrics to improve (become lower) as a result of eliminating weak voiced portions of the signal in the metric calculations. Likewise, higher unvoiced in error percentages will cause the metrics to degrade (become higher) as a result of including unvoiced segments in the calculations. This can be seen in Table 5.2, which shows the results of the MP GHI when using different threshold for the SVD voiced/unvoiced detector. The absolute deviation mean (adm) per utterance is calculated as

$$adm_j = \frac{1}{N_j}\sum_{i=1}^{N_j}\left| f_{ij} - \hat{f}_{ij} \right|, \tag{5.1}$$

where f_{ij}, and $\hat{f}_{ij}$ is the actual and estimated fundamental frequency respectively for sample i, and N_j is the number of samples in utterance j.

The population absolute deviation mean is expressed as

$$adm = \frac{1}{K}\sum_{j=1}^{K} adm_j , \tag{5.2}$$

Where K is the number of speech signals in the population.

The standard deviation (sd_j) per utterance is expressed as

$$sd_j = \frac{1}{N_j - 1}\sum_{i=1}^{N_j}\left(\left| f_{ij} - \hat{f}_{ij} \right| - adm_j \right)^2 , \tag{5.3}$$

and the p.s.d as

$$p.s.d = \frac{1}{K}\sum_{j=1}^{K} sd_j \ . \tag{5.4}$$

As seen in Table 5.1, the voiced in error is the same for the eSPRD and the MP GHI. The performance is commensurate with prior super-resolution techniques.

Table 5.2:
Fundamental estimation evaluation for male speech (top) and female speech (bottom) with varying thresholds for the SVD voiced unvoiced detector.

MP GHI SVD Threshold	Unvoiced in error (%)	Voiced in error (%)	Gross high errors (%)	Gross low errors (%)	Absolute deviation (Hz) mean	Absolute deviation (Hz) p.s.d.
1.0	12.32	2.05	1.51	2.36	2.09	2.84
1.5	8.77	4.10	1.44	1.93	2.01	2.74
2.0	6.10	6.40	1.33	1.60	1.91	2.59
2.5	4.15	9.14	1.00	1.34	1.80	2.41
2.75	3.64	10.55	0.97	1.26	1.74	2.31
3.0	2.92	11.80	0.94	1.08	1.68	2.17
3.03	2.88	12.08	0.96	1.07	1.67	2.16
3.1	2.74	12.65	0.98	0.92	1.65	2.11
3.2	2.57	13.2	0.98	0.94	1.64	2.09
1.0	2.50	4.24	2.02	0.59	3.51	4.91
1.5	1.57	5.79	1.43	0.47	3.32	4.55
2.0	1.11	7.6	1.28	0.29	3.14	4.17
2.25	0.96	8.65	1.12	0.25	3.05	3.97
2.33	0.89	9.06	1.09	0.22	3.02	3.91
2.38	0.84	9.31	1.08	0.23	3.00	3.87
2.5	0.75	9.88	1.03	0.16	2.94	3.70

The gross error, the metric used in [51], represents outliers of the estimated fundamental frequency. This metric does not measure the method's resolution. Gross high errors are the percentage of estimated fundamental frequencies that are 20% greater than the actual. Likewise, gross low errors are the percentage of estimated fundamental frequencies that are 20% lower than the actual. As seen in Table 5.1, the MP GHI is commensurate with the eSRPD and SRPD methods in the gross error metric.

The last results that are worth examining are based on perfect detection results. Perfect detection is when the voice/unvoiced detector correctly classifies every frame. Unfortunately, [49] does not examine this test. As seen in Table 5.3, the results are very good. The male results are very impressive; whereas the female's gross high has increased significantly. This may be a concern, but without eSPRD results to compare to, this is left as an open question. Enhancements to GHI as described in chapter 6, may further improve results in future work.

Table 5.3: MP GHI fundamental estimation evaluation during perfect detection.

Gender	Unvoiced in error (%)	Voiced in error (%)	Gross high errors (%)	Gross low errors (%)	Absolute deviation (Hz) mean	Absolute deviation (Hz) p.s.d.
male	0	0	3.57	2.50	2.15	2.91
female	0	0	9.67	1.05	3.79	5.46

5.6 Conclusions

The MP GHI process is a novel approach to estimating, tracking and assessing the fundamental and harmonic frequencies of a speech signal. Pitch Estimation is a key step in accomplishing many speech processing applications, such as automated speaker identification, speech data compression, pitch alteration, pitch prediction and natural sounding time compressions and expansions. The GHI could also be applied to other types of signals that are periodic in nature.

The GHI process is computationally efficient in that it consists of a small number of trivial matrix calculations and comparisons. In many signal processing applications, signal decomposition such as the MP technique may already be required.

Therefore, the computational efficiency of the GHI process can be easily leveraged by these signal decomposition processes. Also, the GHI process can be implemented as a real-time process, in that an output fundamental and harmonic estimate can be generated for each signal segment, without the need to wait for future segments to be processed.

The GHI process is not confined to any particular super-resolution signal decomposition, but is particularly suited to the MP technique due to its ability to pre-condition the decomposition, based on decay or growth rates, frequencies, initial phases, and initial amplitudes. The GHI allows for super-resolution tracking of both the fundamental and harmonics. It does not require a fundamental component to actually be present in the original signal, since the fundamental candidates are generated based on the spacing between frequency components. A variety of outputs are provided including average fundamental, α_0, harmonic assessment count, c, refined fundamental and harmonic estimates, all of which can be more useful as a group as opposed to methods that simply yield the fundamental estimate itself. Tracking is enhanced as a result of incorporating the estimates of fundamental and harmonics from the previous signal segment. Finally, because the GHI process uses the super-resolution list, $\vec{F}$, for refinement, the output harmonic estimates, h_k, can be used to assess inharmonicity. Inharmonicity occurs when the harmonics are not exact integer multiples of the fundamental, and can be fairly common for example in musical instruments.

The GHI has many advantages as discussed. These advantages, in addition to the results that were observed, demonstrate that the MP GHI is a very attractive pitch estimator.

Chapter 6

Conclusions

6.1 Conclusions and Future Work

In this dissertation it was shown that the MP algorithm can be applied to speech processing problems. In chapter 2 the mathematical theory of the MP algorithm was described. The model of the exponentially damped or un-damped sinusoidal model (ESM) was shown and how the MP approach is applied to a speech signal. The problems with numerical stability, and solutions to these types of problems were presented. The estimation of the speech reconstruction error using the MP algorithm was also introduced. In chapter 3, the MP was applied to the speech compression problem. Following this in chapter 4, techniques used to enhance speech via the MP algorithm were presented. A tone detection technique was introduced and several tone removal techniques were introduced. A new method called the Band Focus Matrix Pencil Temporal Reconstruction (BFMP-TR) addressed the removal of unwanted tones in a speech signal. In chapter 5, a voiced/ unvoiced detection scheme was introduced, and the Generalized Harmonicity Indicator (GHI) was introduced. The GHI is a pitch estimation algorithm that estimates the pitch of an individual's speech.

The MP algorithm is a useful, pre-existing algorithm for decomposition of ESM signals. It has benefits in both directional finding applications and, as in this dissertation,

speech processing applications. It was shown that the MP algorithm can be applied to speech compression, speech enhancement, and pitch estimation. The benefits of the MP algorithm over other estimation techniques, include super resolution of the spectrum, low variance of the parameters are approaching the Cramer-Rao bound, and the fact that it is a nonstochastic process which uses a direct data approach to obtain the model's parameters. The drawback to the MP algorithm is that it is a computational time consuming process. However, it was shown that speech compression can run at a rate of 5 to 7 times slower than real time, using Matlab on a Pentium 4 machine running Windows XP. This indicates that the MP algorithm has a clear potential running in real time when implemented on a Digital Signal Processor (DSP) board.

The ESM can be leveraged in other ways than what was shown in chapter 3. In [14], the authors describe a new type of vocoder that is based on an all-pole model of the vocal tract. This method determines the closest signal, based on a L_2 norm that satisfies the all-pole model. The only parameters that would need to be transmitted are the model's parameters and initial value for each frame. The MP algorithm, also based on an all pole model, could be a potential algorithm used to enhance this vocoder. Rather than transmitting prediction coefficients and initial conditions which can lead to reconstruction instabilities, stable estimates of parameters obtained from application of the MP algorithm can be transmitted.

Another interesting topic that the MP algorithm can be useful for is speech compression using a psycho-acoustic model. The research in [1] looked at this type of speech coder by applying the Total Least Square (TLS) algorithm to the ESM. In this coder the MPEG 1 – Layer 1 psycho-acoustic model was applied. This allowed each sub-

band to be modeled according to its perceptual relevance. An appealing research topic would be to look at the relationship between the MP algorithm and the TLS algorithm. A comparison of computational speed and accuracy results would be interesting. Also, implementing this speech compression technique on a real time DSP would be of interest to the speech processing community.

A MP based filter was developed in chapter 4, called the band focus MP with temporal reconstruction (BFMP-TR). It is a filter that removes unwanted tonal interferes from a speech signal, while removing little speech information. This high resolution technique allows the tone to be more readily separated from the speech. The BFMP-TR algorithm was compared to a typical notch filter. The BFMP-TR group delay is much smaller than the notch filter's delay with equivalent or better results. This would allow the BFMP-TR algorithm to be used in real time systems.

In chapter 5 a new method was presented for fundamental frequency tracking. Referred to as the Generalized Harmonicity Indicator (GHI), this new method was applied to the pitch tracking problem for speech signals.

The GHI has potential to be a more robust pitch estimator than it already is. With regard to the pre-processing that has been described, one could also pre-condition the input frequency list based on phase and decay groupings. This may eliminate many of the frequencies that are not harmonic in nature. Another possible alteration is to search for other sub-harmonics (such as one-third of the fundamental or one-fourth of the fundamental) in the Sub-harmonic Searcher. This would be important for example when certain harmonics of the fundamental are not present in the signal and therefore the difference between harmonics is a non-unity integer multiple of the fundamental. Also,

mathematical models for inharmonicity have been developed and can be used to aid in the search when inharmonicity is potentially present. One could also consider using more than a single delay element on the outputs of the Fundamental Refiner and the Harmonic Refiner to allow for further refinement based on past segments. Furthermore, one could consider non-real time applications where advance elements would allow for refinements based on future segments, in addition to past segments. Finally with respect to the voiced unvoiced detector, more research needs to be done in using the SVD as a detector. Investigations could consider averages of the singular values, or looking at a correlation between frames to determine if it is voiced or unvoiced. Regardless of all this potential work the GHI performs very well with respect to the comparisons in the results. The potential future work will only improve its performance.

In this research, a selected set of important applications have been implemented with the MP. There are other speech processing applications that could benefit from the MP algorithm. One such application is co-channel interference mitigation. Co-channel interference occurs when there are two speakers on one channel at the same time. This problem is a difficult problem to solve, and currently there is no viable solution. The MP algorithm could potentially help solve this problem by decomposing the speech signal into components, determine which components belong to each speaker, and then recombine the components for the appropriate speakers.

Another application that can benefit from the MP algorithm is SID. Currently, SID utilizes the cepstrum to separate the glottal pulse from the vocal tract. Although the cepstrum based SID algorithm performs well, the problem occurs when the

speakers are used to train a SID system under different channel conditions than testing. An example of this would be training the speaker on an RF channel, and testing him or her on a telephone channel. Since the human is able to identify a speaker under different channel conditions, the information must be present in the signal. Therefore, there should be a better method to perform SID. Utilizing the GHI algorithm, a SID algorithm could be developed by analyzing the pitch information. One method that may be successful is to analyze the different moments of the pitch signal. The pitch information would not change under many practical channel conditions. Therefore a GHI SID algorithm would allow the algorithm to perform well, independently of the channel.

Voice transformation is another application that can benefit from the MP algorithm. Voice transformation is an application of speech processing that changes a speaker's voice to sound like another person. The first step in the voice transformation process is to estimate the pitch frequency and the harmonic frequencies. Secondly, the pitch frequency and its harmonic frequencies need to be shifted. The last step is to reconstruct the speech. The GHI could be used to estimate the pitch, and the MP algorithm could be used to shift the pitch. This can be accomplished by shifting the appropriate poles in the z-plane. Finally, once the poles are shifted, the signal could be reconstructed with the shifted MP parameters.

This research displayed a variety of important speech processing topics that benefited from the MP technique including compression, interference removal and pitch estimation. The MP algorithm may also be used in other applications other than speech processing, radio-directional finding, high resolution imaging of moving

targets, analysis of complex modes in lossless closed conduction structures, analysis of propagation of signals over perforated ground planes, and many others [28]. Any area that uses an ESM could benefit from the MP algorithm. A few areas include SONAR, Radar, and Seismic applications. The MP algorithm could help improve some of the applications in these areas, as it did for speech compression, speech enhancement, and pitch estimation.

Appendix A

The TIMIT corpus of read speech is designed to provide speech data for acoustic-phonetic studies and for the development and evaluation of automatic speech recognition systems. TIMIT contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. The TIMIT corpus includes time-aligned orthographic, phonetic and word transcriptions as well as a 16-bit, 16kHz speech waveform file for each utterance [38].

A list of speech files from the TIMIT Acoustic-Phonetic Continuous Speech Corpus that were used to test the compression and BFMP-TR algorithm is as follows. The speech data consist of 190 audio signals, 38 speakers, speaking 5 sentences each.

faks0_sa1.wav
faks0_sa2.wav
faks0_si1573.wav
faks0_si2203.wav
faks0_si943.wav
fcmr0_sa1.wav
fcmr0_sa2.wav
fcmr0_si1105.wav
fcmr0_si1735.wav
fcmr0_si475.wav
fdac1_sa1.wav
fdac1_sa2.wav
fdac1_si1474.wav
fdac1_si2104.wav
fdac1_si844.wav
fdrd1_sa1.wav
fdrd1_sa2.wav
fdrd1_si1544.wav
fdrd1_si1566.wav
fdrd1_si2149.wav
felc0_sa1.wav
felc0_sa2.wav
felc0_si1386.wav
felc0_si2016.wav
felc0_si756.wav
fjas0_sa1.wav
fjas0_sa2.wav
fjas0_si1400.wav
fjas0_si2030.wav
fjas0_si770.wav
fjem0_sa1.wav
fjre0_si1746.wav
fjwb0_sa1.wav
fjwb0_sa2.wav
fjwb0_si1265.wav
fjwb0_si635.wav
fjwb0_si992.wav
fpas0_sa1.wav
fpas0_sa2.wav
fpas0_si1272.wav
fpas0_si2204.wav
fpas0_si944.wav
fram1_sa1.wav
fram1_sa2.wav
fram1_si1360.wav
fram1_si522.wav
fram1_si730.wav
fslb1_sa1.wav
fslb1_sa2.wav
fslb1_si1904.wav
fslb1_si644.wav
fslb1_si891.wav
fjem0_sa2.wav
fjem0_si1264.wav
fjem0_si1894.wav
fjem0_si634.wav
fjre0_sa1.wav
fjre0_sa2.wav
fjre0_si1116.wav
fjre0_si1587.wav
mabw0_sa1.wav
mabw0_sa2.wav
mabw0_si1230.wav
mabw0_si1664.wav
mabw0_si2294.wav
mbjk0_sa1.wav
mbjk0_sa2.wav
mbjk0_si1175.wav
mbjk0_si2128.wav
mbjk0_si545.wav
mccs0_sa1.wav
mccs0_sa2.wav
mccs0_si1469.wav
mccs0_si2099.wav
mccs0_si839.wav
mcem0_sa1.wav
mcem0_sa2.wav
mcem0_si1398.wav
mcem0_si2028.wav
mcem0_si768.wav
mdab0_sa1.wav
mdab0_sa2.wav
mdab0_si1039.wav
mdab0_si1669.wav
mdab0_si2299.wav
mdbb0_sa1.wav
mdbb0_sa2.wav
mdbb0_si1195.wav
mdbb0_si1825.wav
mdbb0_si565.wav
mdld0_sa1.wav
mdld0_sa2.wav
mdld0_si1543.wav
mdld0_si2173.wav
mdld0_si913.wav
mgwt0_sa1.wav
mgwt0_sa2.wav
mgwt0_si1539.wav
mgwt0_si2169.wav
mgwt0_si909.wav
mhpg0_sa1.wav
mhpg0_sa2.wav
mhpg0_si1090.wav
mhpg0_si1720.wav
mhpg0_si460.wav
mjar0_sa1.wav
mjar0_sa2.wav
mjar0_si1988.wav
mjar0_si2247.wav
mjar0_si728.wav
mjsw0_sa1.wav
mjsw0_sa2.wav
mjsw0_si1010.wav
mjsw0_si1640.wav
mjsw0_si2270.wav
mmdb1_sa1.wav
mmdb1_sa2.wav
mmdb1_si1625.wav
mmdb1_si2255.wav
mmdb1_si995.wav
mmdm2_sa1.wav
mmdm2_sa2.wav
mmdm2_si1452.wav
mmdm2_si1555.wav

mmdm2_si2082.wav
mpdf0_sa1.wav
mpdf0_sa2.wav
mpdf0_si1542.wav
mpdf0_si2172.wav
mpdf0_si912.wav
mpgl0_sa1.wav
mpgl0_sa2.wav
mpgl0_si1099.wav
mpgl0_si1729.wav
mpgl0_si469.wav
mrcz0_sa1.wav
mrcz0_sa2.wav
mrcz0_si1541.wav
mrcz0_si2171.wav
mrcz0_si911.wav
mreb0_sa1.wav
mreb0_sa2.wav
mreb0_si1375.wav
mreb0_si2005.wav
mreb0_si745.wav
mrgg0_sa1.wav
mrgg0_sa2.wav
mrgg0_si1199.wav
mrgg0_si1829.wav
mrgg0_si569.wav
mrjo0_sa1.wav
mrjo0_sa2.wav
mrjo0_si1364.wav
mrjo0_si1624.wav
mrjo0_si734.wav
msjs1_sa1.wav
msjs1_si639.wav
msjs1_si869.wav
mstk0_sa1.wav
mstk0_sa2.wav
mstk0_si1024.wav
mstk0_si2222.wav
mstk0_si2284.wav
mtas1_sa1.wav
mtas1_sa2.wav
mtas1_si1473.wav
mtas1_si2098.wav
mtas1_si838.wav
mtmr0_sa1.wav
mtmr0_sa2.wav
mtmr0_si1303.wav
mtmr0_si1933.wav
mtmr0_si673.wav
mwbt0_sa1.wav
mwbt0_sa2.wav
mwbt0_si1553.wav
mwbt0_si2183.wav
mwbt0_si923.wav
mwew0_sa1.wav
mwew0_sa2.wav
mwew0_si1361.wav
mwew0_si1991.wav
mwew0_si731.wav
mwvw0_sa1.wav
mwvw0_sa2.wav
mwvw0_si1476.wav
mwvw0_si2106.wav
mwvw0_si846.wav

Reference

[1] Hermus, K.; Verhelst, W.; Wambacq, P.;"Psychoacoustic modeling of audio with exponentially damped sinusoids"; IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings, 2002.; Volume 2, Page(s):1821 - 1824

[2] Tufts, D.W.; Kumaresan, R; "Estimation of frequencies of multiple sinusoids: Making linear prediction perform like maximum likelihood"; Proceedings of the IEEE Volume 70, Issue 9, Sept. 1982 Page(s):975 - 989

[3] Kumaresan, R.; Tufts, D.;"Estimating the parameters of exponentially damped sinusoids and pole-zero modeling in noise"; Acoustics, Speech, and Signal Processing Volume 30, Issue 6, Dec 1982 Page(s):833 - 840

[4] Paulraj, A.; Roy, R.; Kailath, T.; "A subspace rotation approach to signal parameter estimation"; Proceedings of the IEEE; Volume 74, Issue 7, July 1986 Page(s):1044 – 1046

[5] Roy, R.; Kailath, T.; "ESPRIT-estimation of signal parameters via rotational invariance techniques"; IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume 37, Issue 7, July 1989 Page(s):984 – 995

[6] Strobach, P.; "Fast Recursive Subspace Adaptive ESPRIT Algorithms" IEEE Transactions on Signal Processing, Volume 46, Issue 9, Sept. 1998 Page(s):2413 - 2430

[7] Kung, S.Y.; Arun K.S.; Bhaskar Rao, D.V.; "State-space and singular-value decomposition-based approximation methods for the harmonic retrieval problem"; Jounrnal of Optical Society of America; Vol. 73, No. 12; December 1983; pages 1799-1811

[8] Ouibrahim, H. "A Generalized Approach to Direction Finding," PhD dissertation, Syracuse University, Syracuse, NY Dec. 1986

[9] Ouibrahim, H., Weiner, D. D. and Sarkar, T. K., "A Generalized Approach to Direction Finding," IEEE Transactions on Acoustics, Speech & Signal Processing, Vol. 36, No. 4, pp. 610-612, Apr. 1988.

[10] Hua Y. and Sarkar, T.K.; "Further analysis of three modern techniques for pole retrieval from data sequence", Proceedings 30th Midwest Symposium Circuit System, Syracuse, NY, Aug 1987

[11] Ouibrahim, H., Weiner, D. D. and Wei, Z.Y.; " Angle of Arrival Estimation using Forward-Back Moving Window"; Proceedings 30th Midwest Symposium Circuit System, Syracuse, NY, Aug 1987, pages 563-566

[12] Hua Y. and Sarkar, T.K.; "Matrix Pencil Method and its Performance", IEEE Transactions on Acoustics, Speech & Signal Processing, Apr. 1988.

[13] Laroche, Jean; "The Use Of The Matrix Pencil Method For Spectrum Analysis Of Musical Signals"; J. Acoust Soc. Am. 94 (4) October 1993

[14] Lemmerling, Philippe; Dologlou, Ioannis; Van Huffel, Sabine; "Speech Compression Based on Exact Modeling and Structured Total Least Norm"; IEEE International Conference on Acoustics, Speech, and Signal Processing; Volume 1, 12-15 May 1998 Page(s):353 - 356

[15] Jensen, J.; Jensen, S.H.; Hansen, E.; "Exponential Sinusoidal Modeling Of Transitional Speech Segments"; 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, Volume 1, 15-19 March 1999 Page(s):473 - 476

[16] Jensen, J.; Jensen, S.H.; Hansen, E.; "Harmonic exponential modeling of transitional speech segments**";** 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing; Volume 3, 5-9 June 2000 Page(s):1439 - 1442

[17] Badeau, R.; David, B.; Richard, G; "Selecting The Modeling Order For The ESPRIT High Resolution Method: An Alternative Approach"; IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004; Volume 2, 17-21 May 2004 Page(s):ii - 1025-8

[18] Badeau, R.; Richard, G.; David, B.; "Fast Adaptive Esprit Algorithm"; 13th IEEE/SP Workshop on Statistical Signal Processing, July 17-20 2005 Page(s):289 - 294

[19] Badeau, R.; David, B.; Richard, G.; "High-Resolution Spectral Analysis Of Mixtures Of Complex Exponentials Modulated By Polynomials**";** IEEE Transactions on Acoustics, Speech, and Signal Processing; Volume 54, Issue 4, April 2006 Page(s):1341 – 1350

[20] Deller J., Hansen J., Proakis J., "Discrete-Time Processing of Speech Signals", New York: IEEE Press,, 2000

[21] Oppenheim, Alan V.; Schafer, Ronald W; "Discrete-Time Signal Processing"; Englewood Cliff, New Jersey; Prentice Hall; 1989; pages 447-448

[22] Quatieri, Thomas F.; "Discrete-Time Speech Signal Processing"; Upper saddle River, NJ; Prentice Hall PTR; 2002, page 6-7

[23] F.R. Gantmacher, “The Theory of Matrices II”, Chelsea Publishing, Rhode Island, 1987

[24] Braham Himed and Donald D. Weiner, “ Application of the Matrix Pencil Approach to Directional Finding”, Rome Laboratory Final technical report RL-TR-91-104, June 1991

[25] G.W. Stewart, “On the sensitivity of the Eigenvalue Problem $A\vec{x} = \lambda B\vec{x}$ ”, SIAM J. Num Anal. 9, 1972, 669-686.

[26] Xuedong Huang, Alex Acero, Hsiao-WuenHon, “Spoken Language Processing”, p.276, 2001 Prentice-Hall Inc. Upper Saddle River, New Jersey 07458

[27] Fernandez Del Rio J.E., and Sarkar T. K.; "Comparison Between the Matrix Pencil Method and the Fourier Transform Technique for High-Resolution Spectral Estimation"; Digital Signal Processing, Vol. 6, No. 2, pp. 108-125, 1996.

[28] Tapan K. Sarkar, Odilon Pereira, “Using the Matrix Pencil Method to Estimate the Parameters of a Sum of Complex Exponentials”, IEEE Antennas and Propagation Magazine, Vol. 37, No. 1, February 1995 pp. 48-55.

[29] Yingbo Hua, Tapan K. Sarkar,”Generalized Pencil-of-Function Method for extracting Poles of an EM System from Its Transient Response”, IEEE Transactions on Antennas and Propagation, Vol.37, No.2, February 1989,p.229-234.

[30] Gene H. Golub, Charles F. Van Loan.; “Matrix Computations” 3rd edition, pp. 81-83, 1996 John Wiley & Sons, Inc., New York

[31] Macon, M.W.; Clements, M.A.; “Sinusoidal modeling and modification of unvoiced speech”; IEEE Transactions on Speech and Audio Processing; Volume 5, Issue 6, Nov. 1997 Page(s):557 – 560

[32] Sarkar, Tapan K.; Wicks, Michael C.; Salazar-Palma, Magdalena; and Bonneau, Robert J.; “Smart Antennas”; pp.88-89; 2003 IEEE Press, John Wiley & Sons, Inc., Hoboken, New Jeresy

[33] Eilouti, H.H.; Abu-El-Haija, A.L.; “Waveform smoothing: analysis and comparisons” Instrumentation and Measurement Technology Conference, 1989. IMTC-89. Conference Record., 6th IEEE, 25-27 April 1989 Page(s):313 – 319

[34] Noble, Ben; Daniel, James W.; “Applied Linear Algebra”, pp. 342-345, 1988 Prentice Hall, Englewood Cliffs, NJ

[35] Haykin, Simon “Applied Linear Adaptive Filter Theory”, 3rd edition, pp. 176-179, 1996 Prentice Hall, Upper Saddle River, NJ

[36] Anton, Howard and Rorres, Chris.; “Elementary Linear Algebra” 8th edition, pp. 278, 2000 John Wiley & Sons, Inc., New York

[37] Wai C. Chu, “Speech Coding Algorithms”, New Jeresy: John Wiley & Sons, Inc., 2003

[38] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1

[39] Darren M. Haddad, Tapan K. Sarkar, and Andrew J. Noga, “Speech Compression using the Matrix Pencil”, IEEE 12th Digital Signal Processing Workshop, September 2006

[40] Thomas F. Quatieri and Robert J. McAulay, “Shape Invariant Time-Scale and Pitch Modification of Speech”, IEEE Trans. On Signal Processing, Vol. 40, No. 3, March 1992

[41] Howard W. Sabrin, “Unix Speech Processing Development”, Rome Laboratory Final Technical Report RL-TR-97-97, October 1997.

[42] Cividino, Lorenzo; “Power factor, harmonic distortion; causes, effects and considerations”; Telecommunications Energy Conference, 1992. INTELEC '92., 14th International 4-8 Oct. 1992 Page(s):506 - 513

[43] Campbell, Joseph P. JR.; “Speaker Recognition: A Tutorial”; Proceedings of the IEEE Vol.85, No.9, September 1997

[44] Ramachandran R.P., Zilovic M.S., Mammone R.J.; “A Comparative Study of Robust Linear Prediction Analysis Methods with Applications to Speaker Identification”, IEEE Transaction on Speech and Audio Processing, pp. 117-125, Vol. 3, No. 2, 1995

[45] Wenndt, S.J., Noga A.J.; “Narrow-Band Interference Cancellation for Enhanced Speaker Identification”, Applications of Signal Processing to Audio and Acoustics, 1999 IEEE Workshop on17-20 Oct. 1999, Page(s):123 – 126

[46] Gold, N. Morgan, “Speech and Audio Signal Processing”, John Wiley & Sons, Inc., 2000.

[47] X. Sun, “Pitch Determination and Voice Quality Analysis Using Subharmonic-to-Harmonic Ratio,” IEEE Conference on Acoustics Speech and Signal Processing, ICASSP’02, 2002.

[48] Y. Medan, E. Yair, D. Chazan, “Super Resolution Pitch Determination of Speech Signals,” IEEE Trans. On Signal Processing, ASSP-39(1):40-48, 1991.

[49] P. Bagshaw, S. Hiller, M. Jack, “Enhanced Pitch Tracking and the Processing of F0 Contours for Computer Aided Intonation Teaching,” 3rd European Conference on Speech Communication and Technology, EUROSPEECH’93, Berlin, Germany, September 1993.

[50] de Cheveigné, Alain; Kawahara, Hideki “YIN, a fundamental frequency estimator for speech and music”, Acoustical Society of America Journal, Volume 111, Issue 4, pp. 1917-1930 (2002).

[51] Bishnu S. Atal, Lawrence R. Rabiner; “A Pattern Recognition Approach to Voiced-Unvoiced-Silence Classification with Applications to Speech Recognition”, IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume ASSP-24, No. 3, June 1976, pp. 201-212.

[52] http://www.cstr.ed.ac.uk/research/projects/fda/

VITA

Name of Author: Darren M. Haddad

Place of Birth: New Hartford, New York

Date of Birth: April 4, 1967

Degrees:

M.S., Electrical Engineering, Syracuse University, May 1999
B.S., Electrical Engineering, Rochester Institution of Technology, August 1991
A.A.S., Electrical Engineering Technology, Mohawk Valley Community College, May 1987

Faculty:

Mathematical adjunct professor, MVCC, 2001

Professional Experience:

- Research Engineer, AFRL, Rome, NY 1994 - present
- Design Engineer, 485 EIG, Griffiss AFB, Rome, NY 1992-1994

Patents:

- Steganographic Method for Covert Audio Communications, Kaliappan Gopalan, Darren M. Haddad, Stanley J. Wenndt (Pending)
- Generalized Harmonic Indicator, Darren M. Haddad, Tappan K. Sarkar, Andy J. Noga (Invention Disclosure)